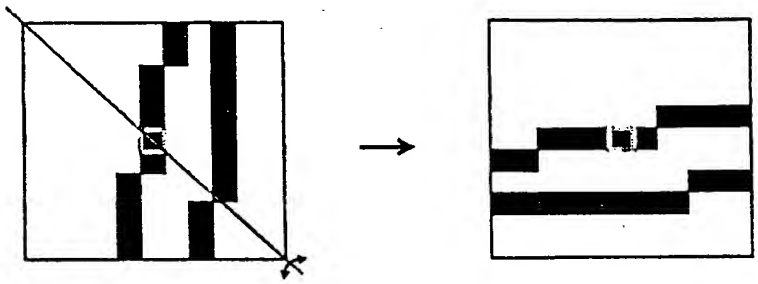


Fig. 1

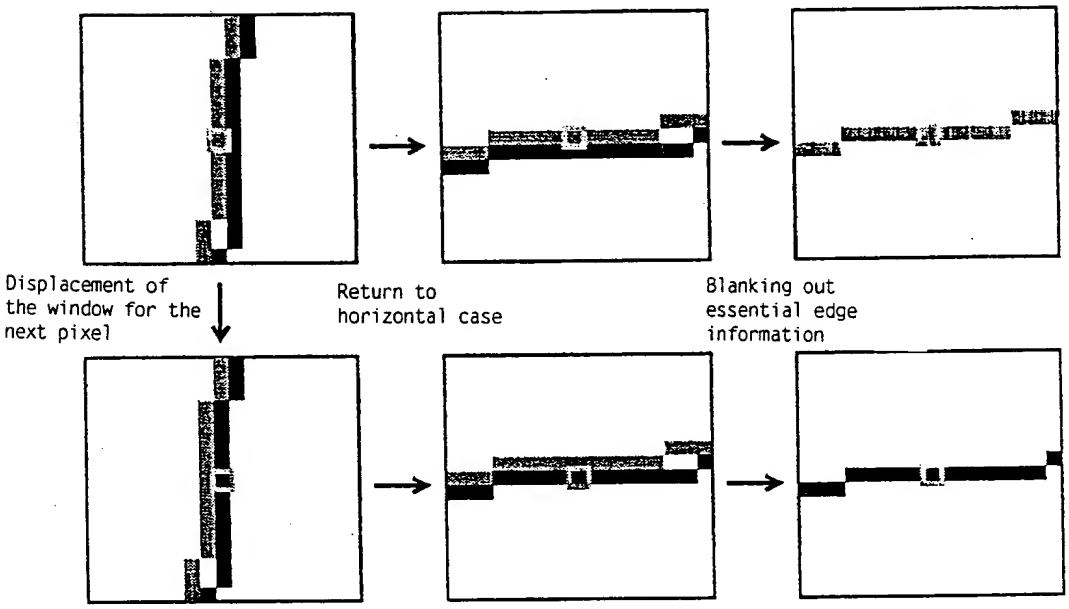


Fig. 2

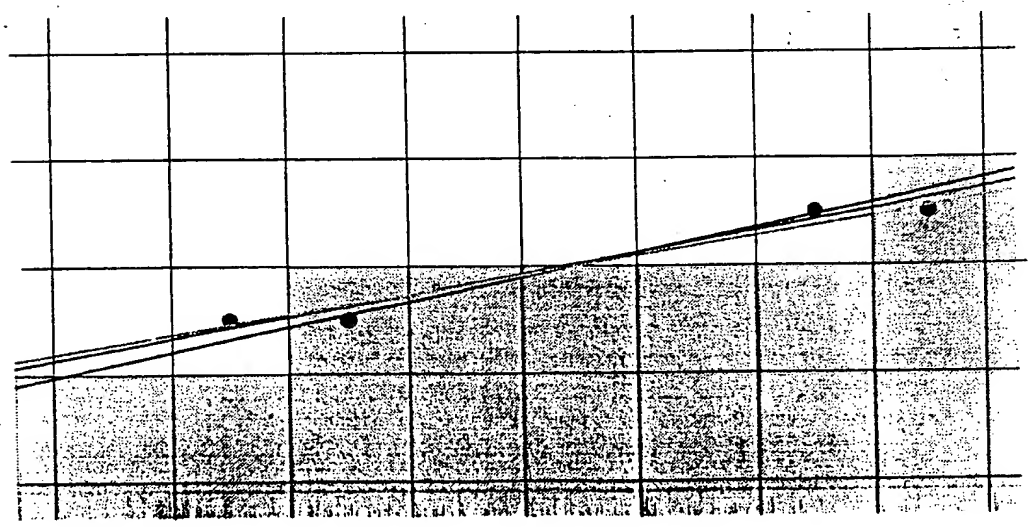


Fig. 3

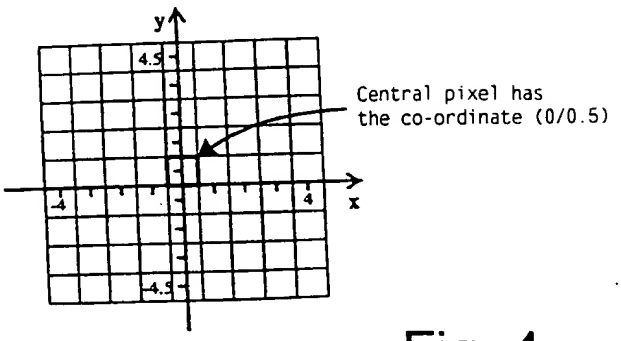
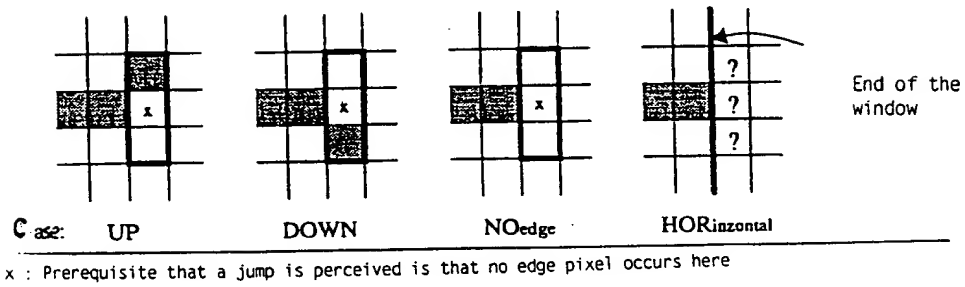


Fig. 4



x : Prerequisite that a jump is perceived is that no edge pixel occurs here

Fig. 5

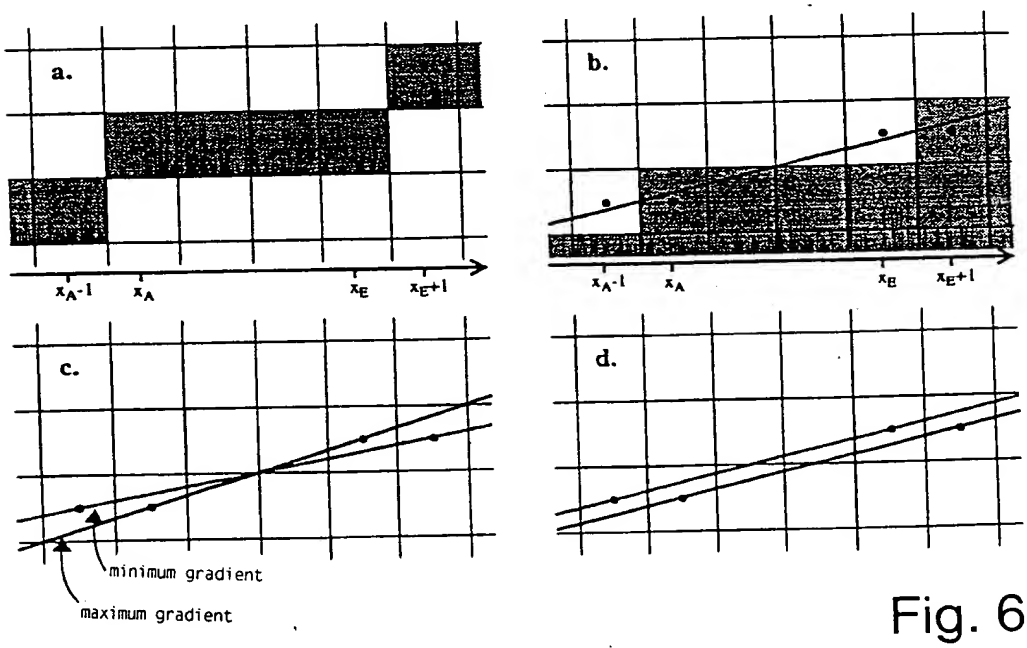


Fig. 6

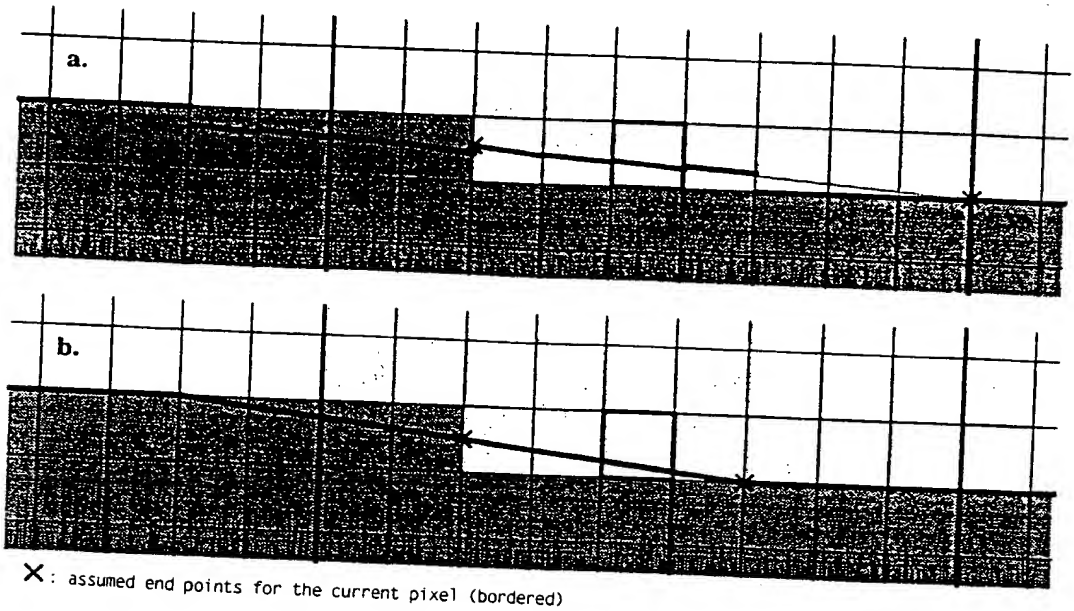

a.
b.
×: assumed end points for the current pixel (bordered)

Fig. 7

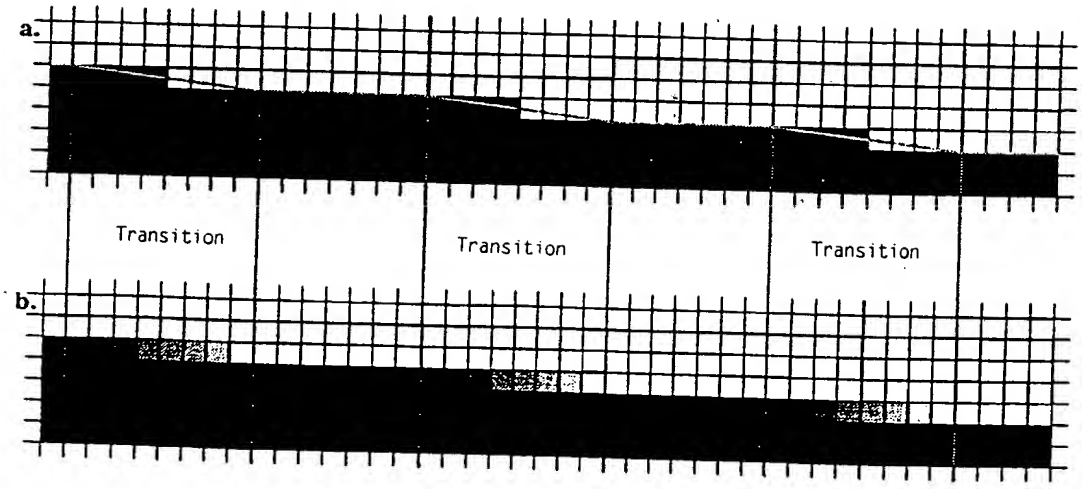
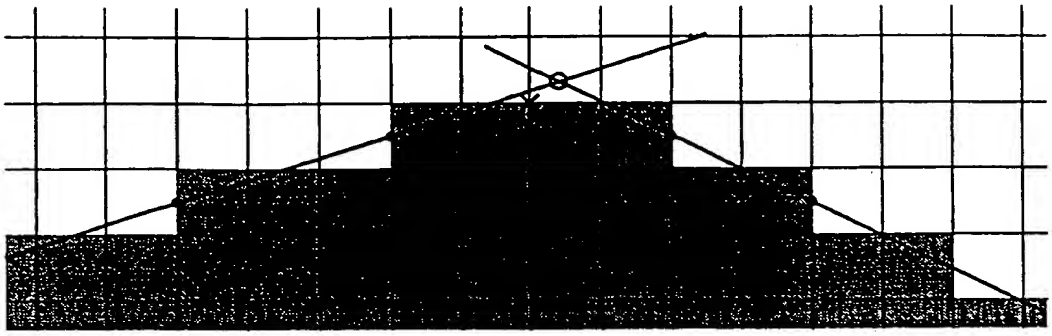

a.
Transition
Transition
Transition
b.

Fig. 8



O :extrapolated center point
X :assumed center point

Fig. 9

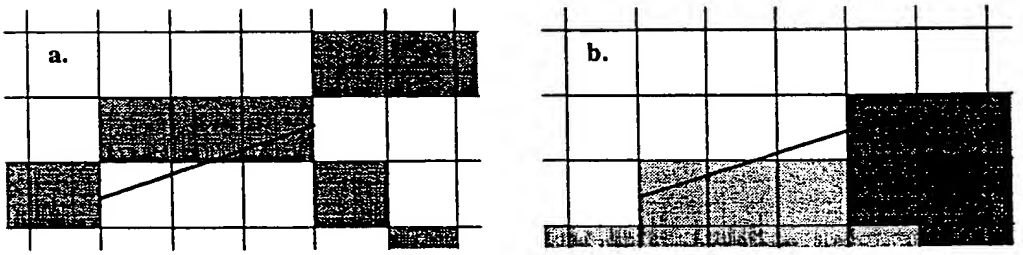


Fig. 10

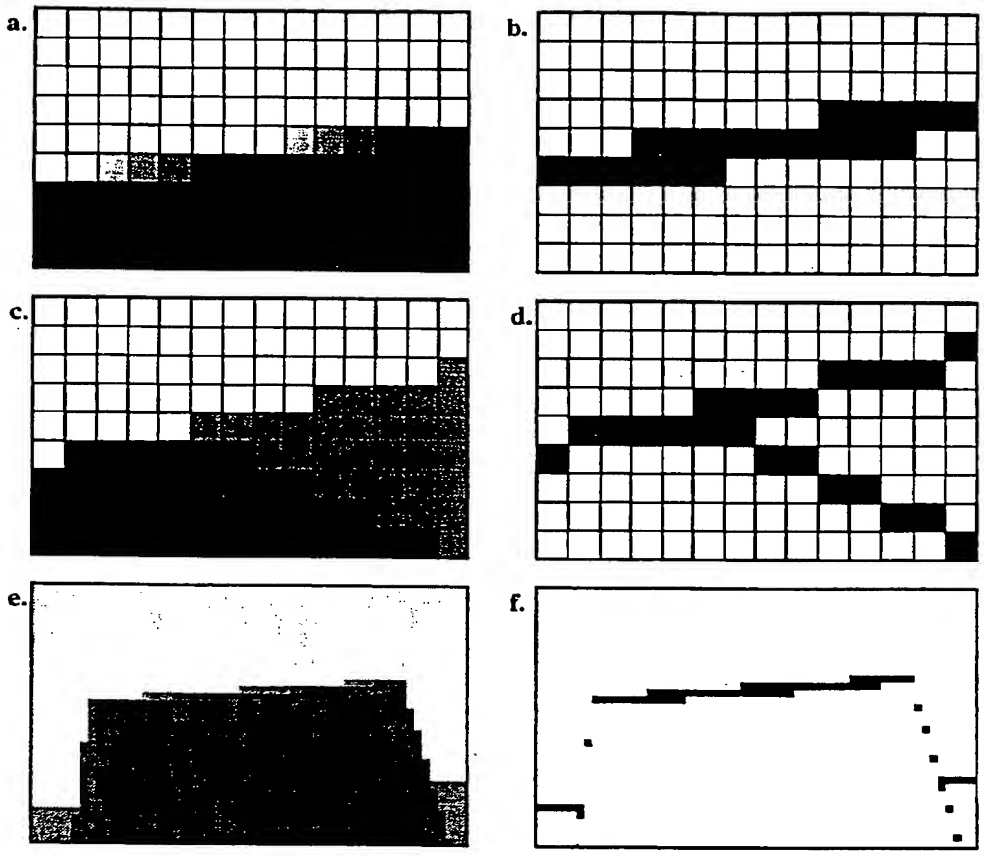


Fig. 11

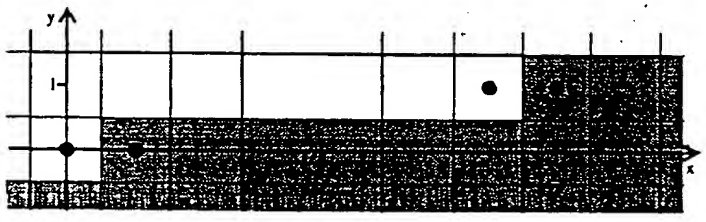


Fig. 12

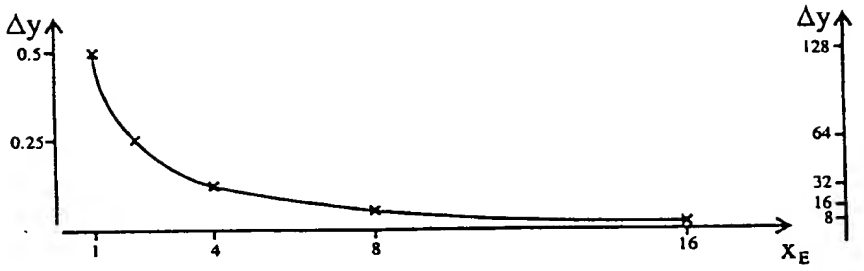


Fig. 13

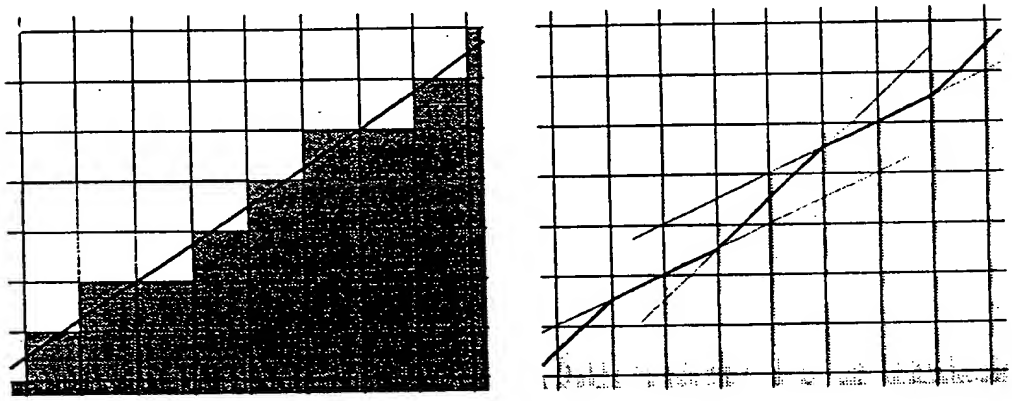


Fig. 14

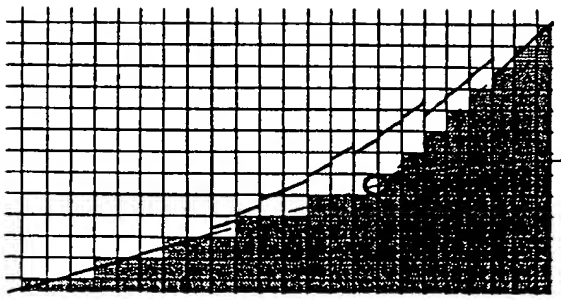


Fig. 16

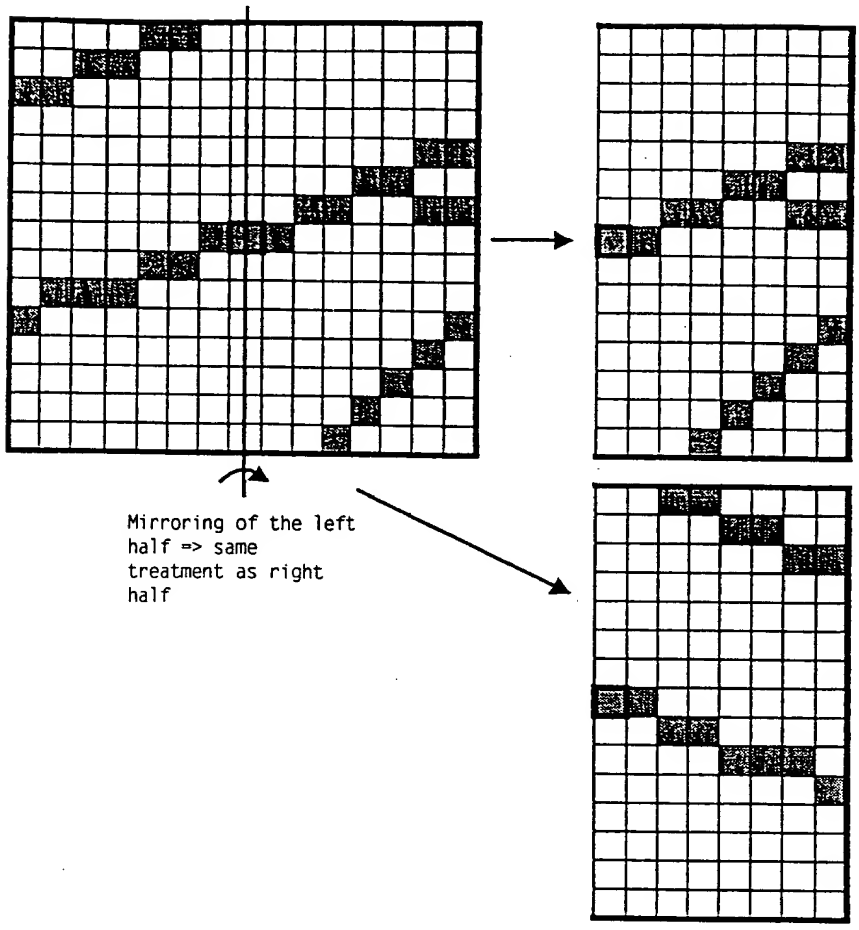

Mirroring of the left half => same treatment as right half

Fig. 15

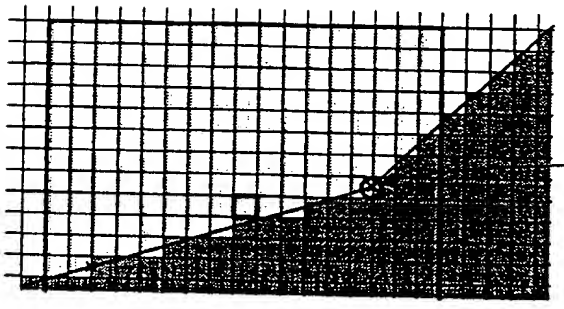

Found corner point at the transition of two edges
× : assumed end points for the current pixel (bordered)

Fig. 17

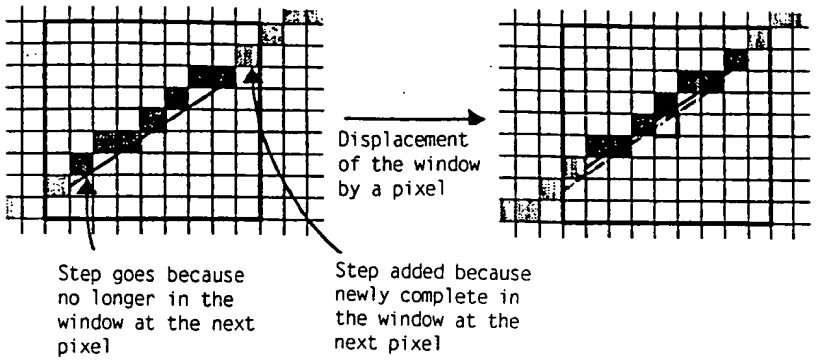


Fig. 18

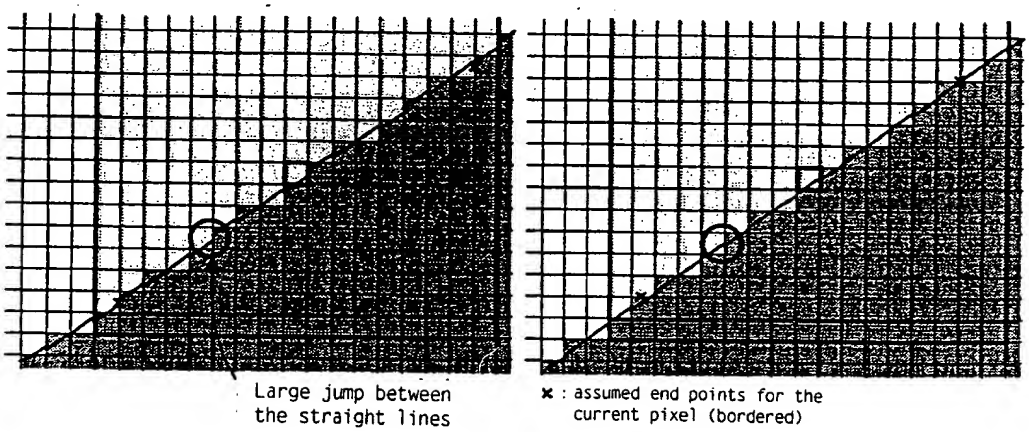


Fig. 19

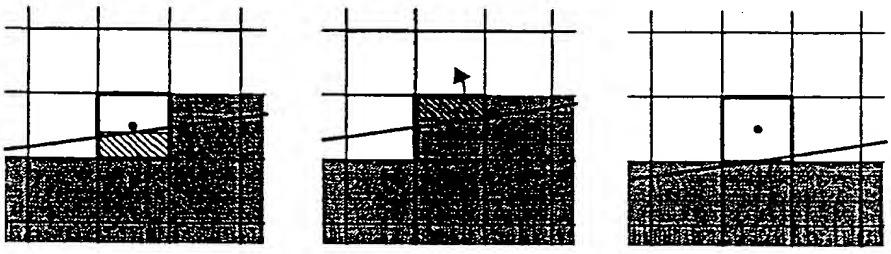
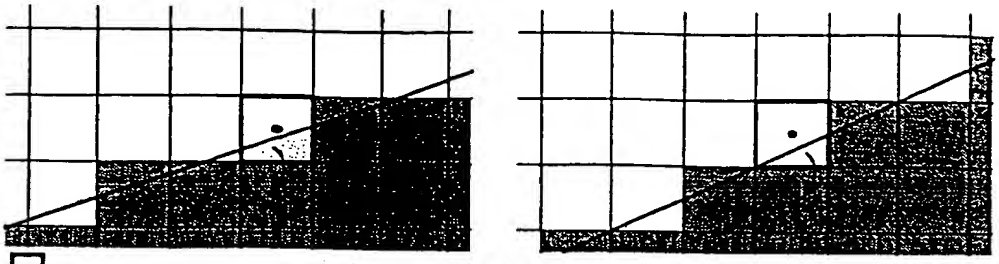


Fig. 20



: current pixel ; : area proportion beneath the straight line and thus also the amount of the color proportion to be mixed from below

Fig. 21

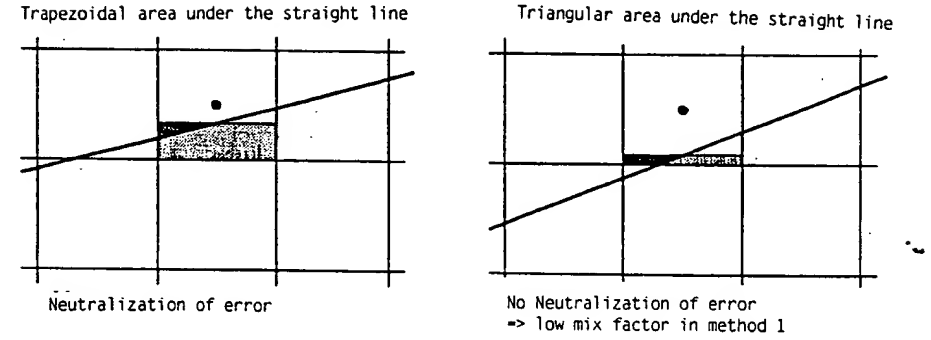


Fig. 22

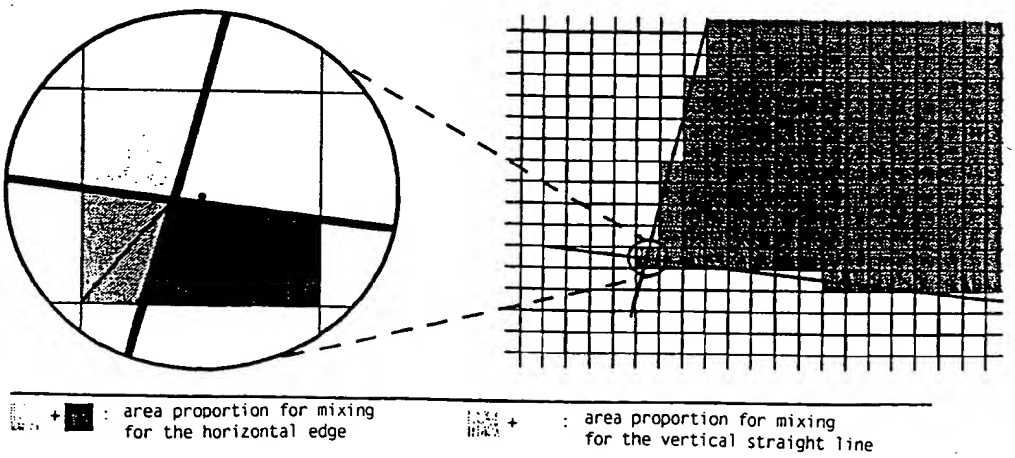


Fig. 23

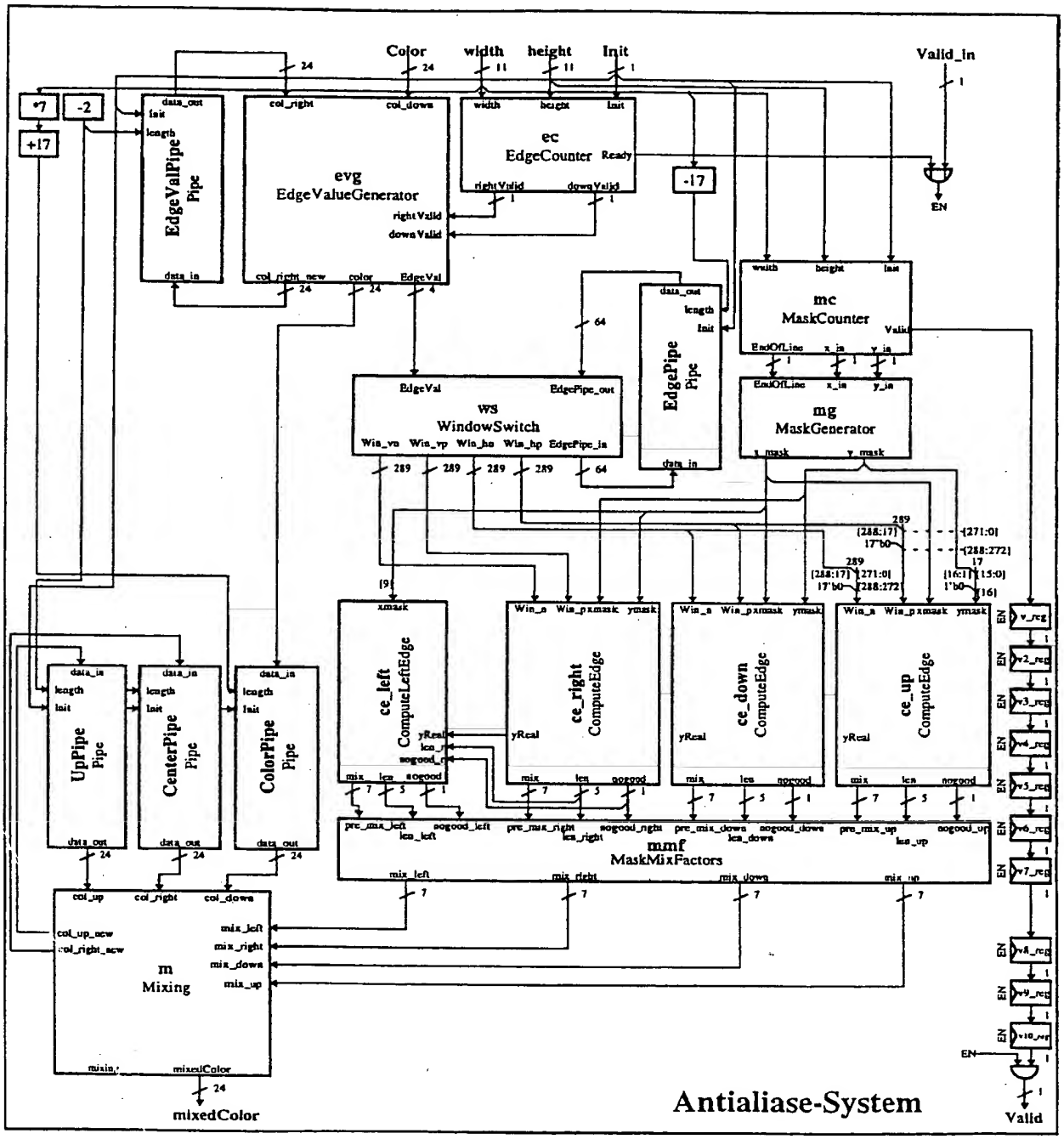

Color
width
height
Init
Valid_In
*7
-2
+17
EdgeValPipe Pipe
evg
EdgeValueGenerator
ec
EdgeCounter
-17
EN
mc
MaskCounter
EdgePipe Pipe
mg
MaskGenerator
WS
WindowSwitch
ce_left
ComputeLeftEdge
ce_right
ComputeEdge
ce_down
ComputeEdge
ce_up
ComputeEdge
UpPipe Pipe
CenterPipe Pipe
ColorPipe Pipe
mmf
MaskMixFactors
m
Mixing
mixedColor
Antialiase-System
Valid

Fig. 24

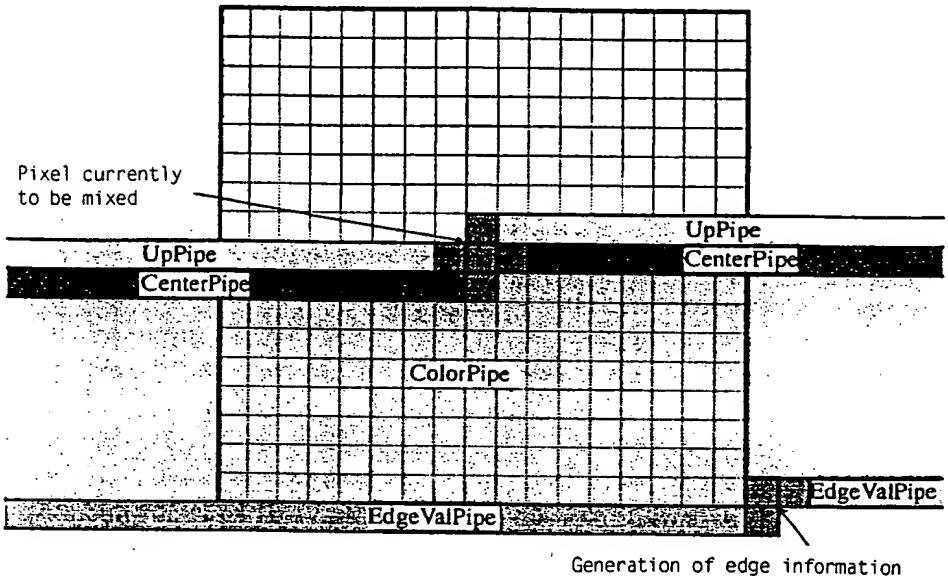

Pixel currently to be mixed
UpPipe
CenterPipe
UpPipe
CenterPipe
ColorPipe
EdgeValPipe
EdgeValPipe
Generation of edge information

Fig. 25

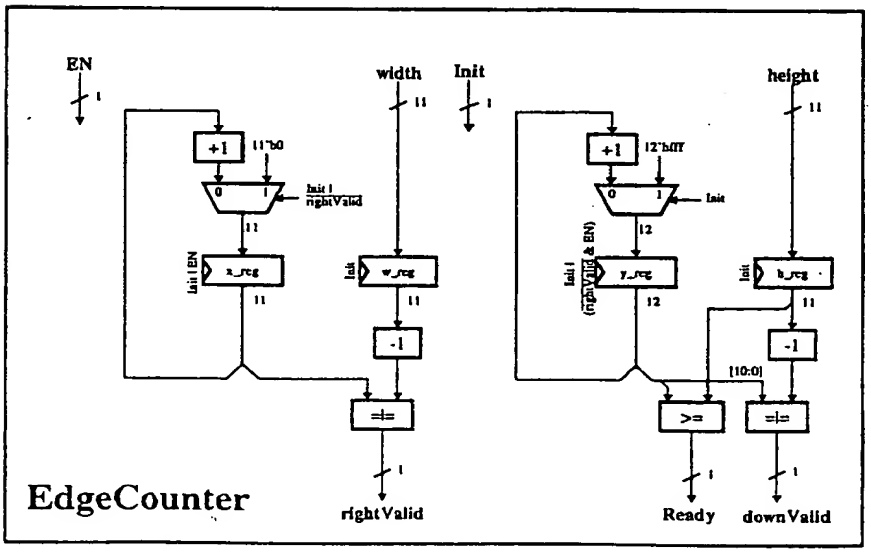

EN
width
Init
height
+1
-1
==
>=
rightValid
Ready
downValid
EdgeCounter

Fig. 26

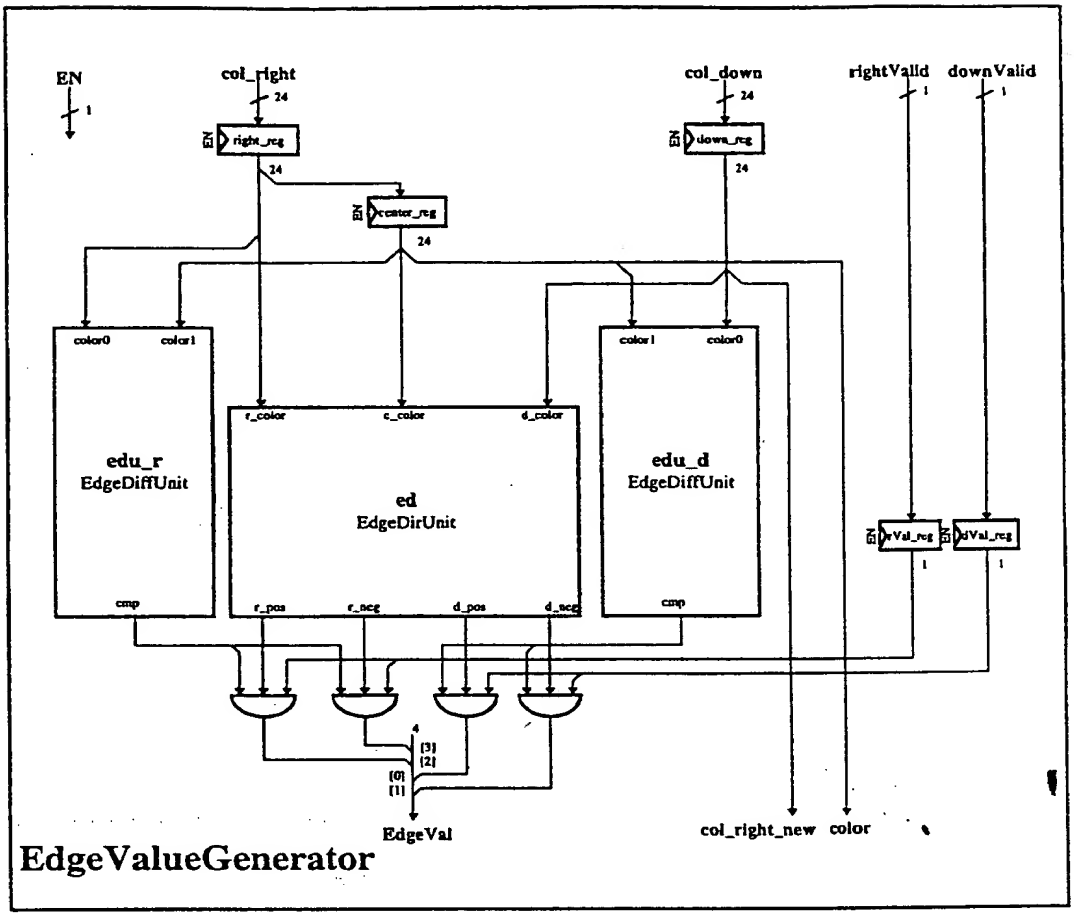

EN
col_right
24
col_down
24
rightValid
downValid
right_reg
down_reg
center_reg
color0
color1
edu_r
EdgeDiffUnit
r_color
c_color
d_color
ed
EdgeDirUnit
edu_d
EdgeDiffUnit
cmp
r_pos
r_neg
d_pos
d_neg
rVal_reg
dVal_reg
EdgeVal
col_right_new
color
EdgeValueGenerator

Fig. 27

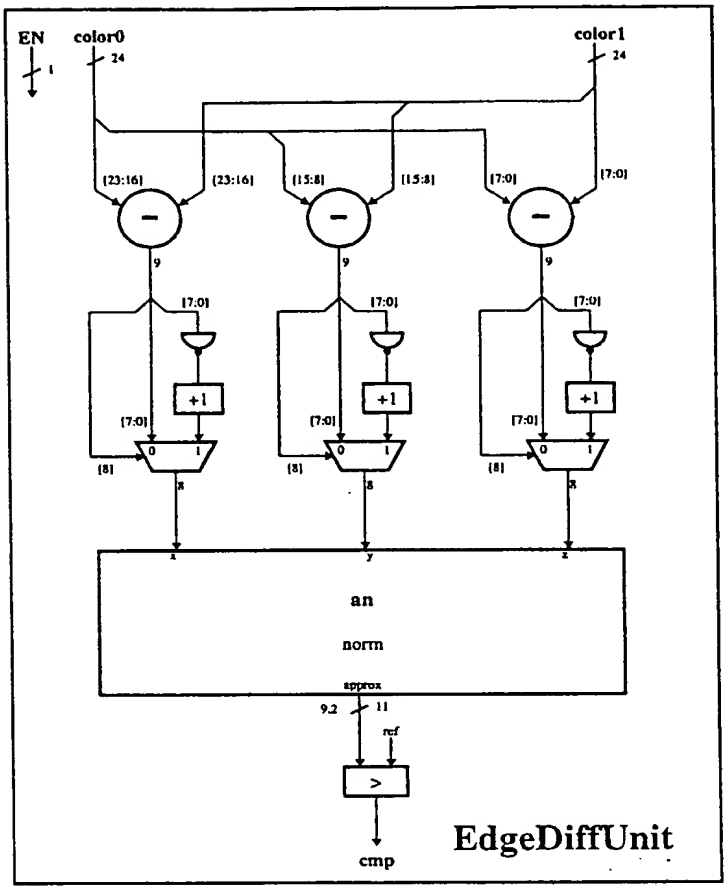

EN
color0
color1
24
[23:16]
[15:8]
[7:0]
9
+1
an
norm
approx
9.2
11
ref
>
cmp
EdgeDiffUnit

Fig. 28

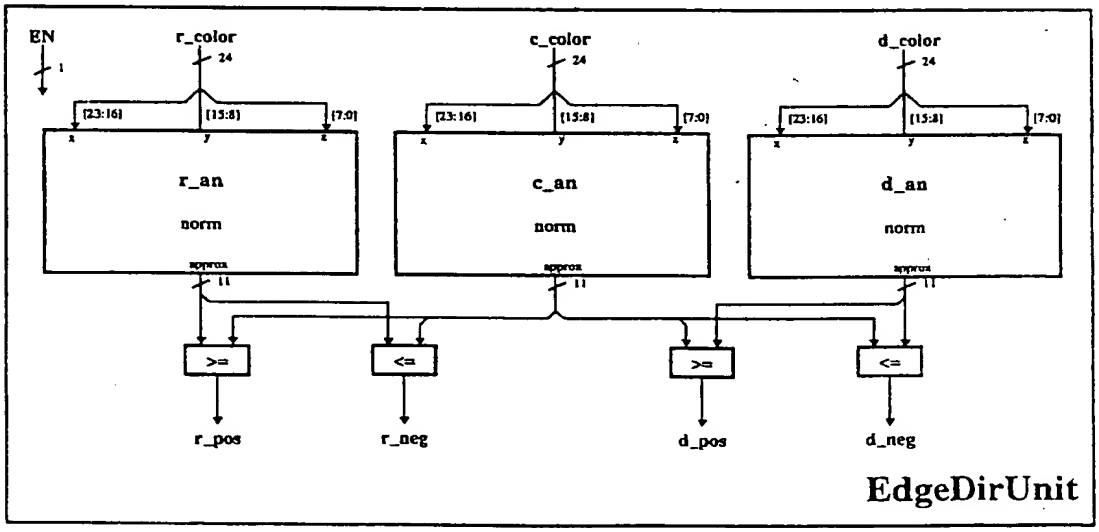

EN
r_color
c_color
d_color
24
[23:16]
[15:8]
[7:0]
r_an
c_an
d_an
norm
approx
11
>=
<=
r_pos
r_neg
d_pos
d_neg
EdgeDirUnit

Fig. 29

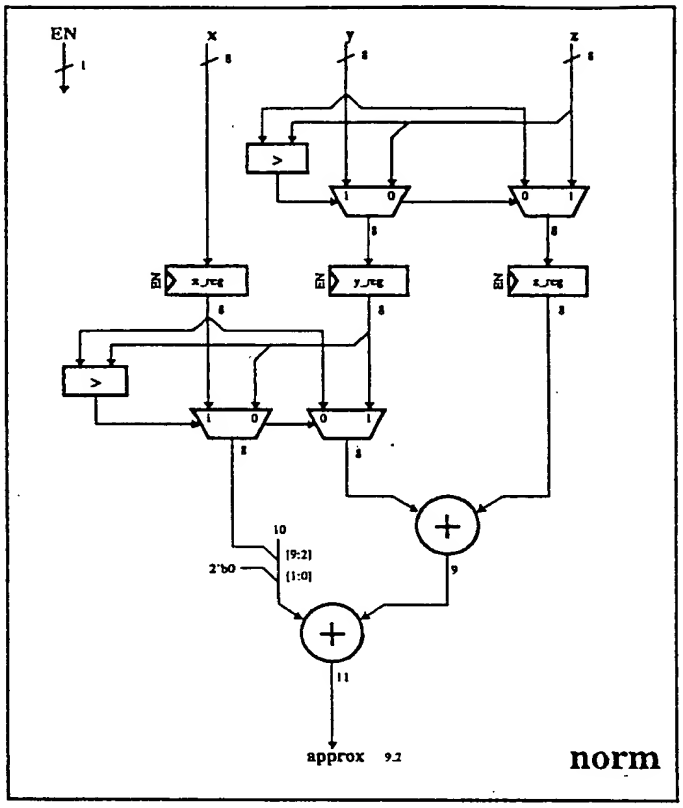

EN
x
y
z
>
EN
x_reg
y_reg
z_reg
2'b0
[9:2]
[1:0]
approx
norm

Fig. 30

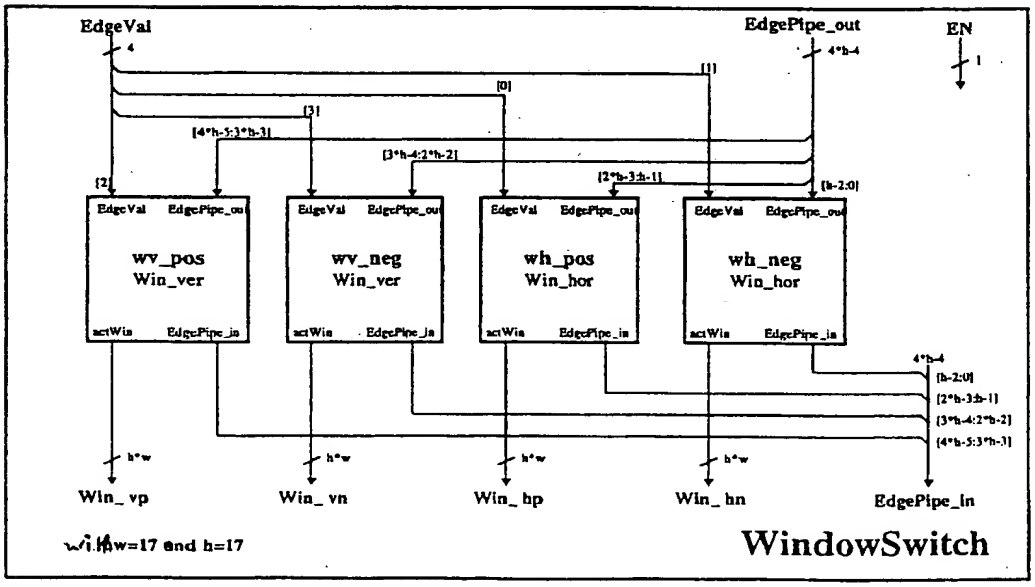
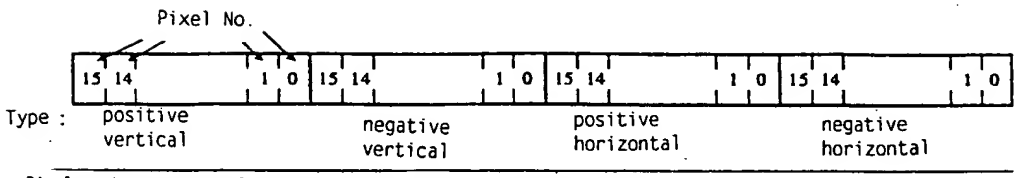

EdgeVal
EdgePipe_out
EN
wv_pos
Win_ver
wv_neg
Win_ver
wh_pos
Win_hor
wh_neg
Win_hor
actWin
EdgePipe_in
Win_ vp
Win_ vn
Win_ hp
Win_ hn
EdgePipe_In
WindowSwitch

Fig. 31



Pixel number 0 is the lowermost pixel in the corresponding column
Pixel number 15 is the uppermost pixel

Fig. 32

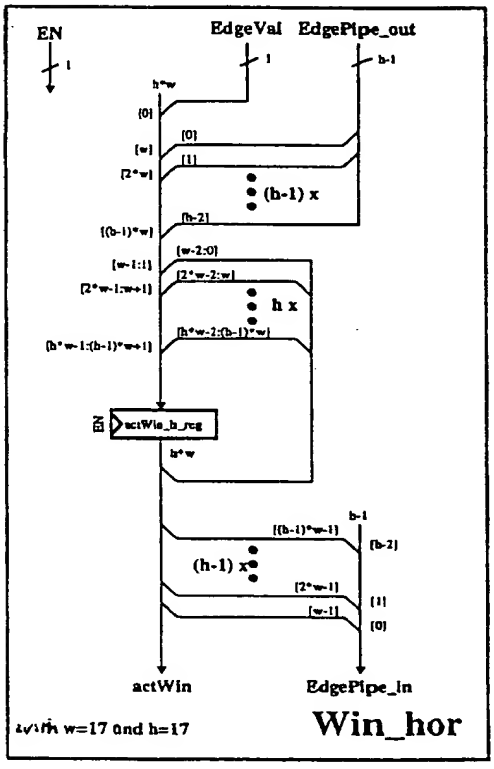


Fig. 33

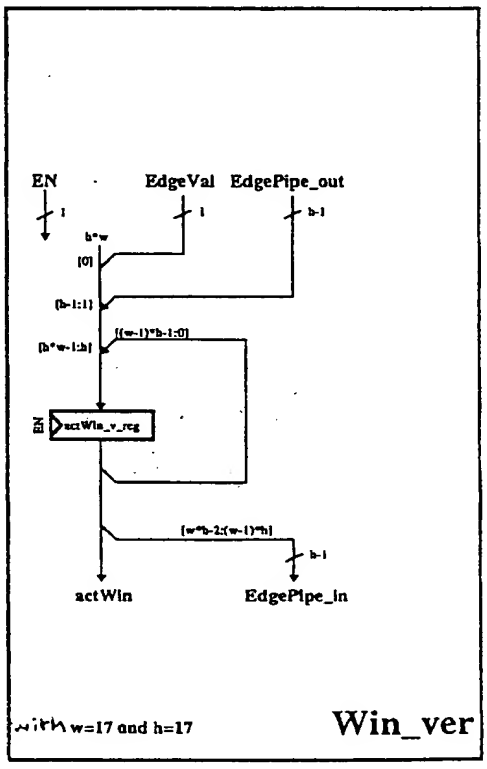


Fig. 34

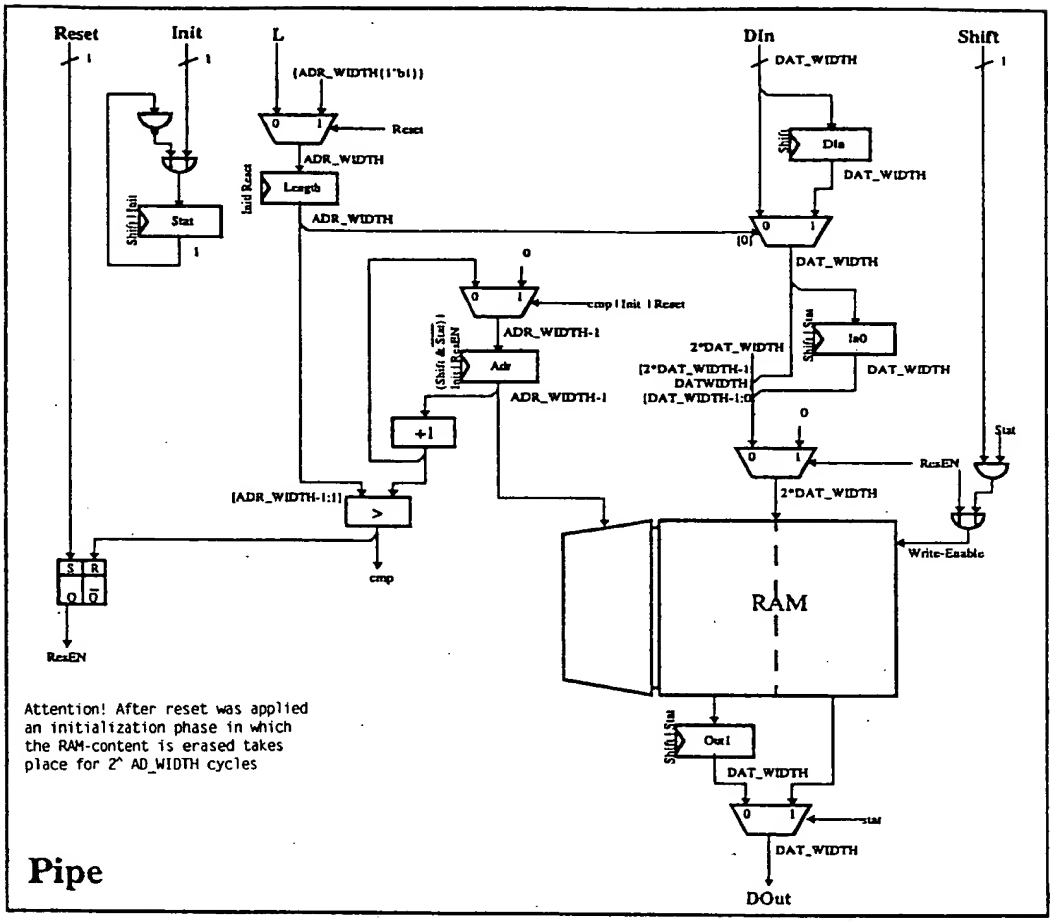


Fig. 35

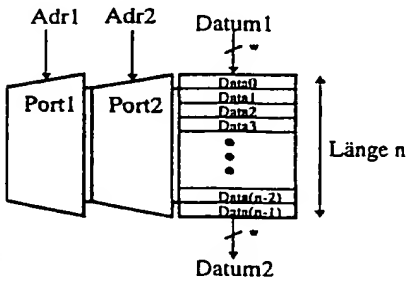


Fig. 36

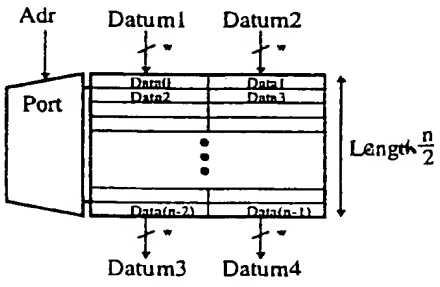


Fig. 37

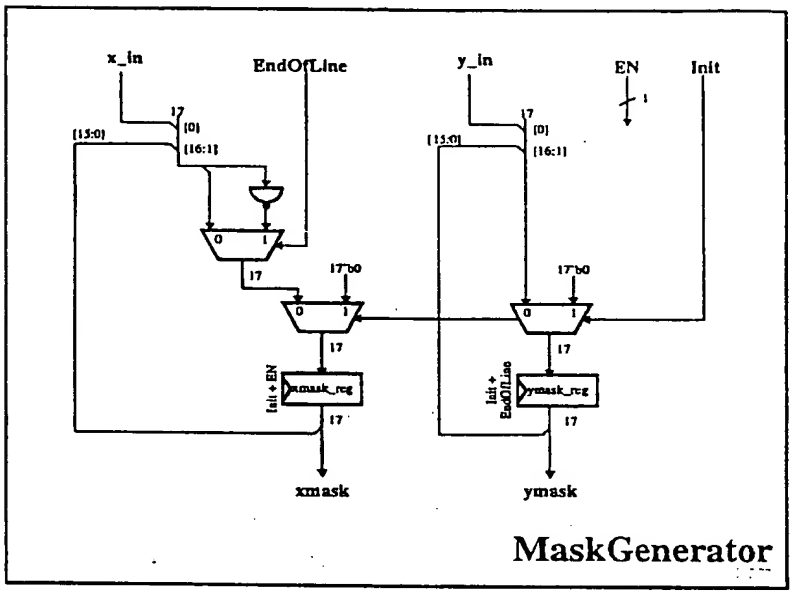

x_in
EndOfLine
y_in
EN
Init
[15:0]
[16:1]
[0]
17
17'b0
xmask_reg
ymask_reg
xmask
ymask
MaskGenerator

Fig. 38

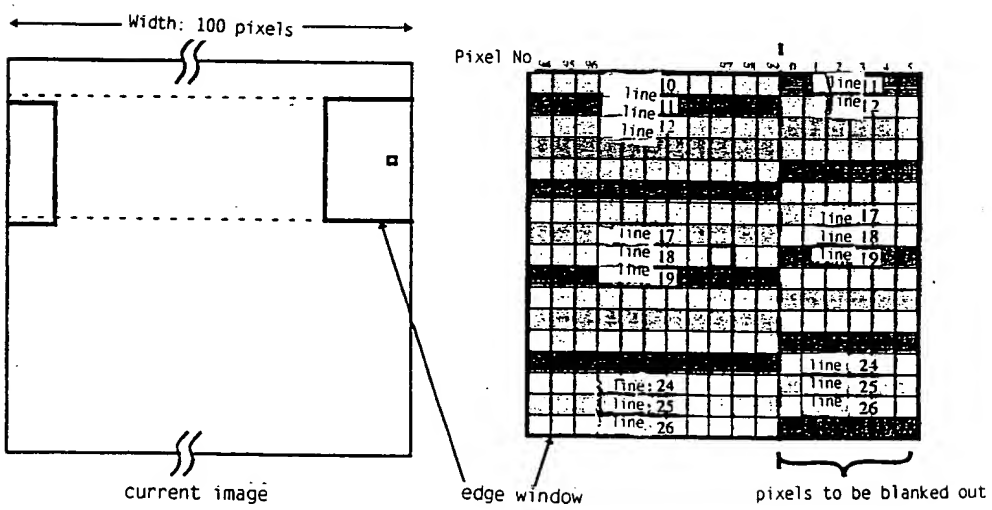

Width: 100 pixels
Pixel No
current image
edge window
pixels to be blanked out

Fig. 39

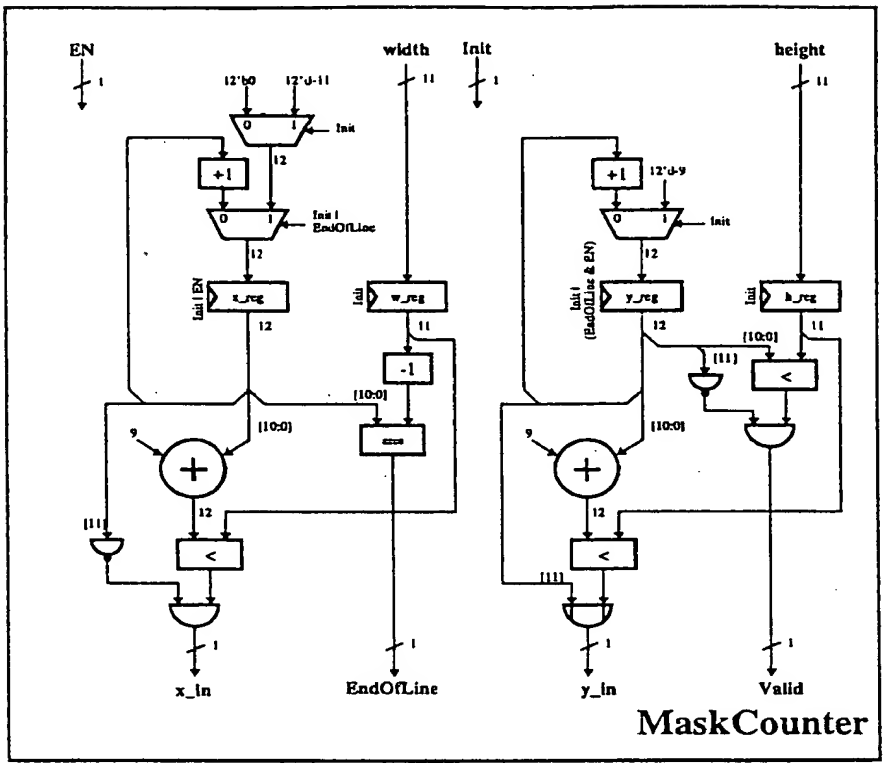

EN
width
Init
height
x_in
EndOfLine
y_in
Valid
MaskCounter

Fig. 40

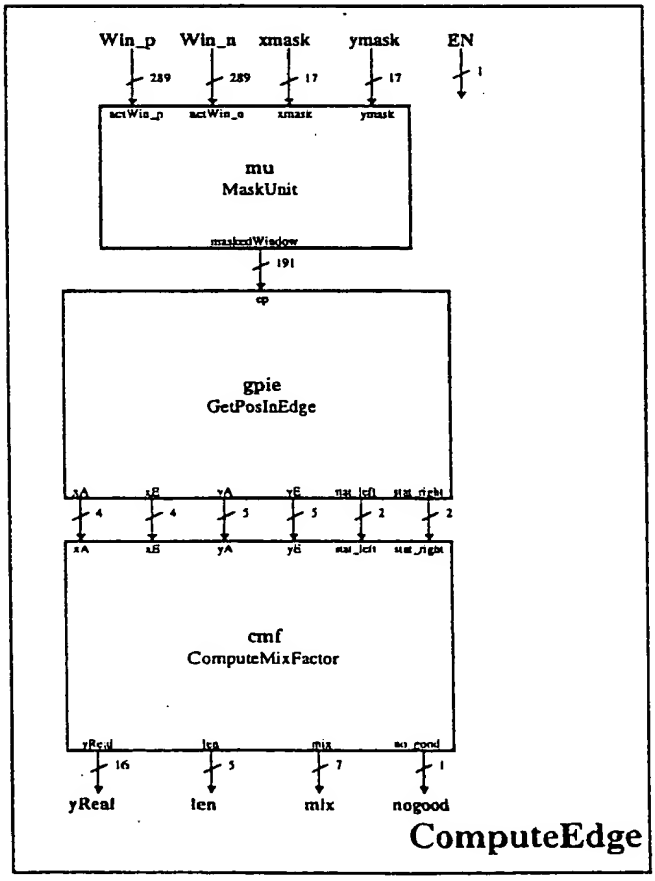

Win_p
Win_n
xmask
ymask
EN
mu
MaskUnit
gpie
GetPosInEdge
cmf
ComputeMixFactor
yReal
len
mix
nogood
ComputeEdge

Fig. 41

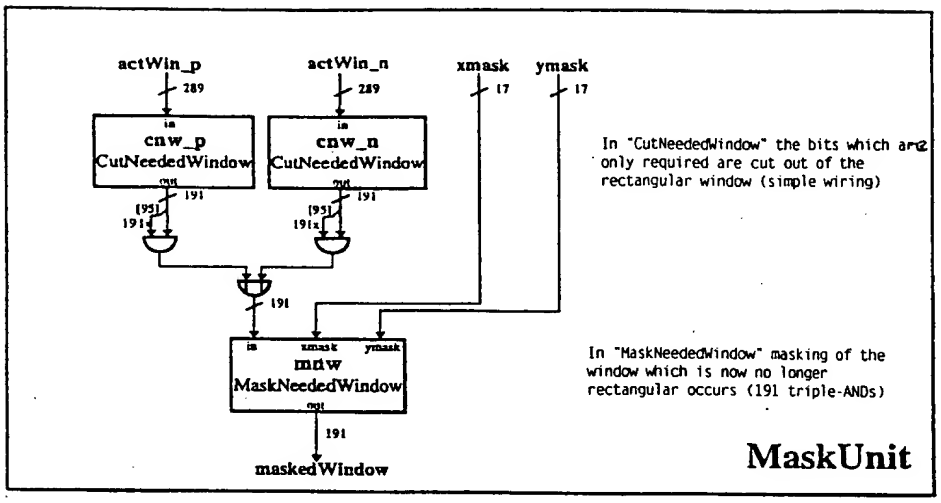

actWin_p
actWin_n
xmask
ymask
289
289
17
17
in
cnw_p
CutNeededWindow
out
cnw_n
CutNeededWindow
191
191
191
191
191
in
xmask
ymask
mnw
MaskNeededWindow
out
191
maskedWindow
In "CutNeededWindow" the bits which are only required are cut out of the rectangular window (simple wiring)
In "MaskNeededWindow" masking of the window which is now no longer rectangular occurs (191 triple-ANDs)
MaskUnit

Fig. 42

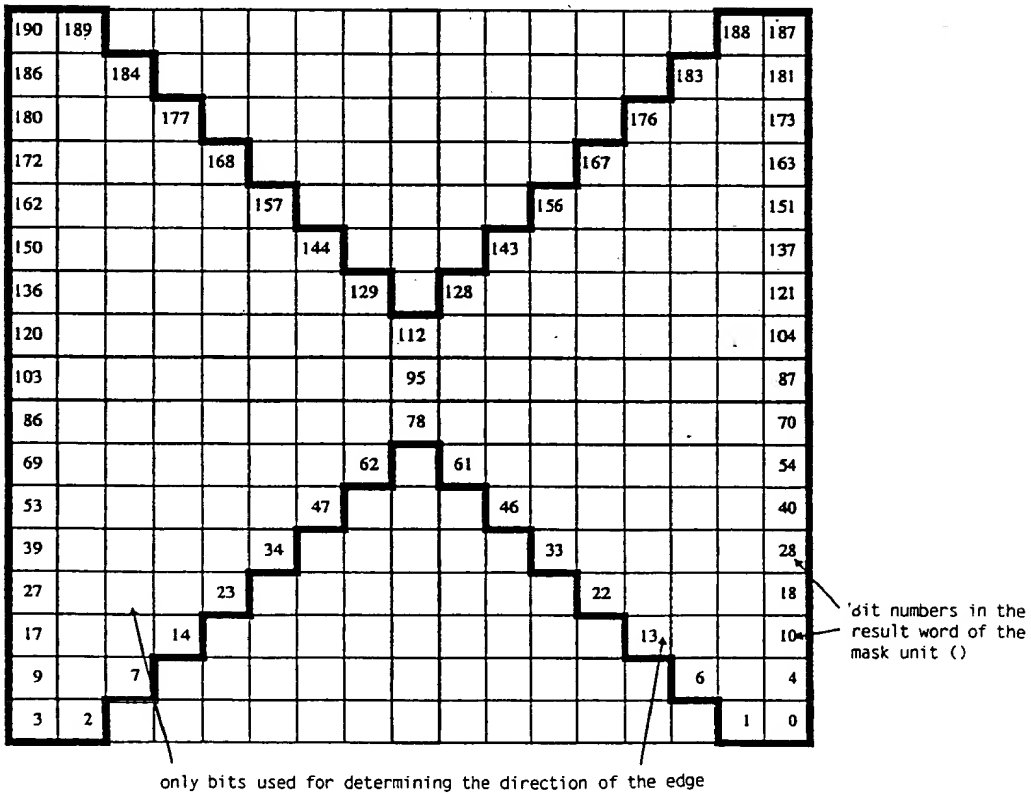

190 189 188 187
186 184 183 181
180 177 176 173
172 168 167 163
162 157 156 151
150 144 143 137
136 129 128 121
120 112 104
103 95 87
86 78 70
69 62 61 54
53 47 46 40
39 34 33 28
27 23 22 18
17 14 13 10
9 7 6 4
3 2 1 0
'dit numbers in the result word of the mask unit ()
only bits used for determining the direction of the edge

Fig. 43

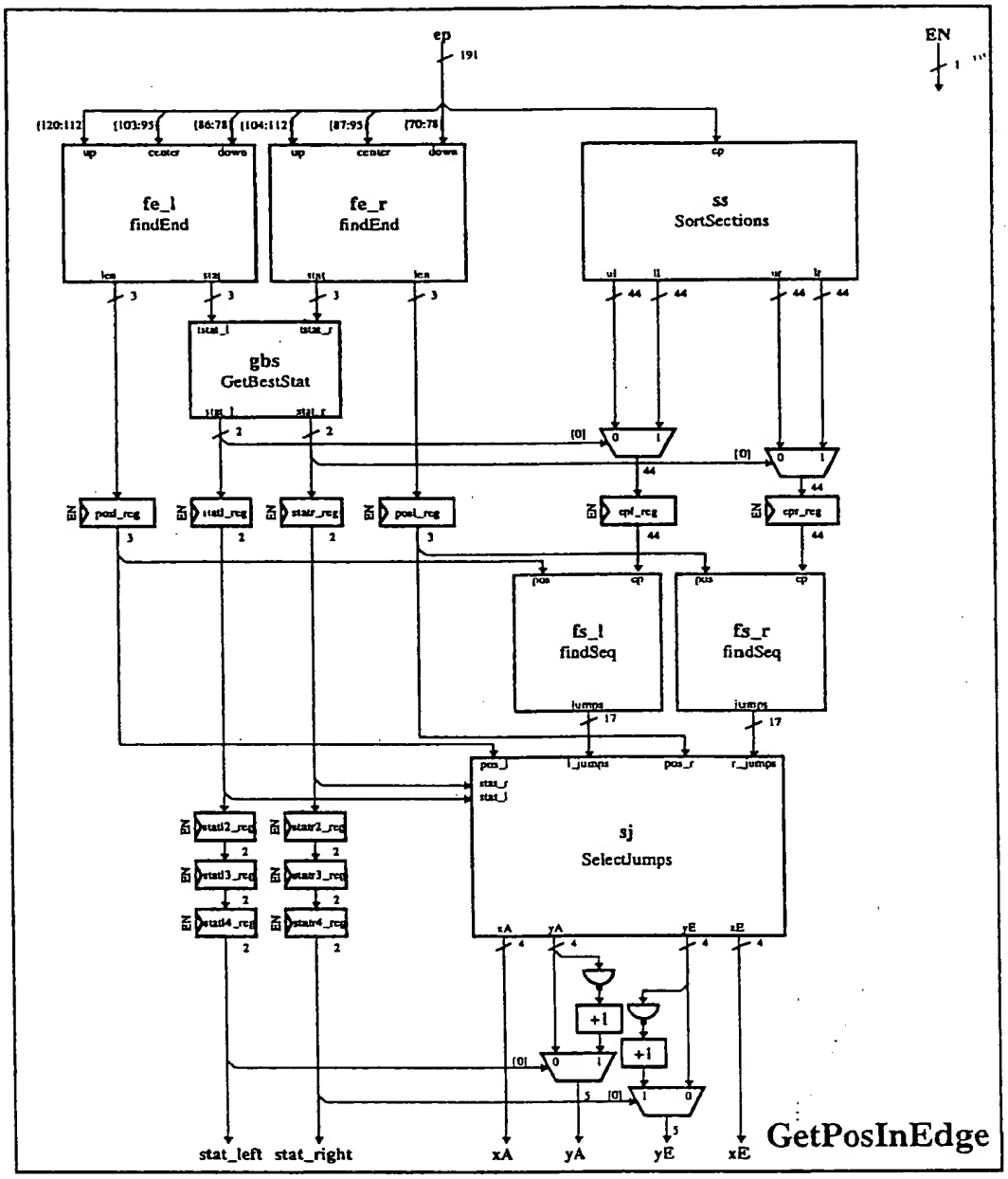

ep
191
EN
fe_l
findEnd
fe_r
findEnd
ss
SortSections
gbs
GetBestStat
fs_l
findSeq
fs_r
findSeq
sj
SelectJumps
stat_left
stat_right
xA
yA
yE
xE
GetPosInEdge

Fig. 44

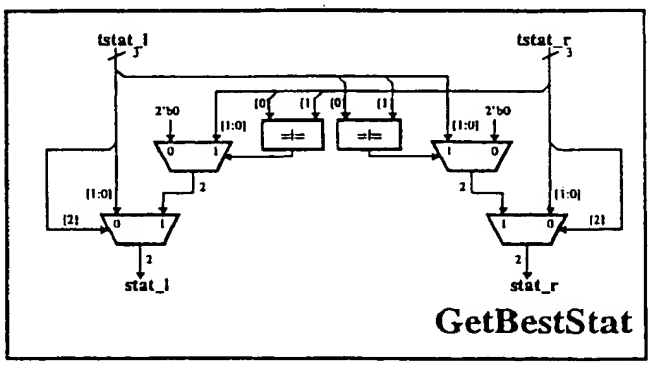

tstat_l
tstat_r
stat_l
stat_r
GetBestStat

Fig. 45

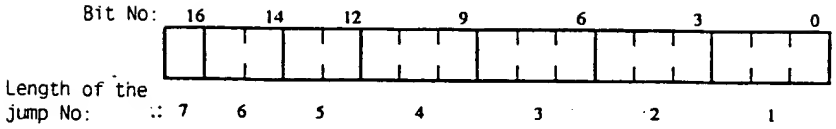

Bit No: 16 14 12 9 6 3 0
Length of the jump No: 7 6 5 4 3 2 1

Fig. 46

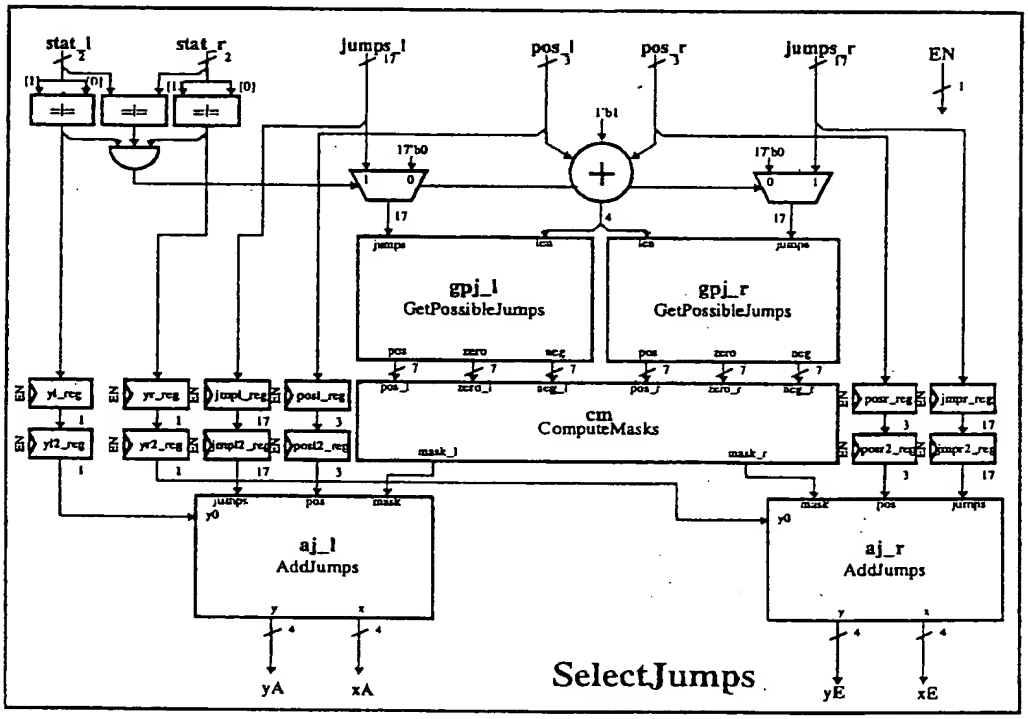

stat_l
stat_r
jumps_l
pos_l
pos_r
jumps_r
EN
17'b1
17'b0
17'b0
gpj_l
GetPossibleJumps
gpj_r
GetPossibleJumps
cm
ComputeMasks
aj_l
AddJumps
aj_r
AddJumps
yA
xA
yE
xE
SelectJumps

Fig. 47

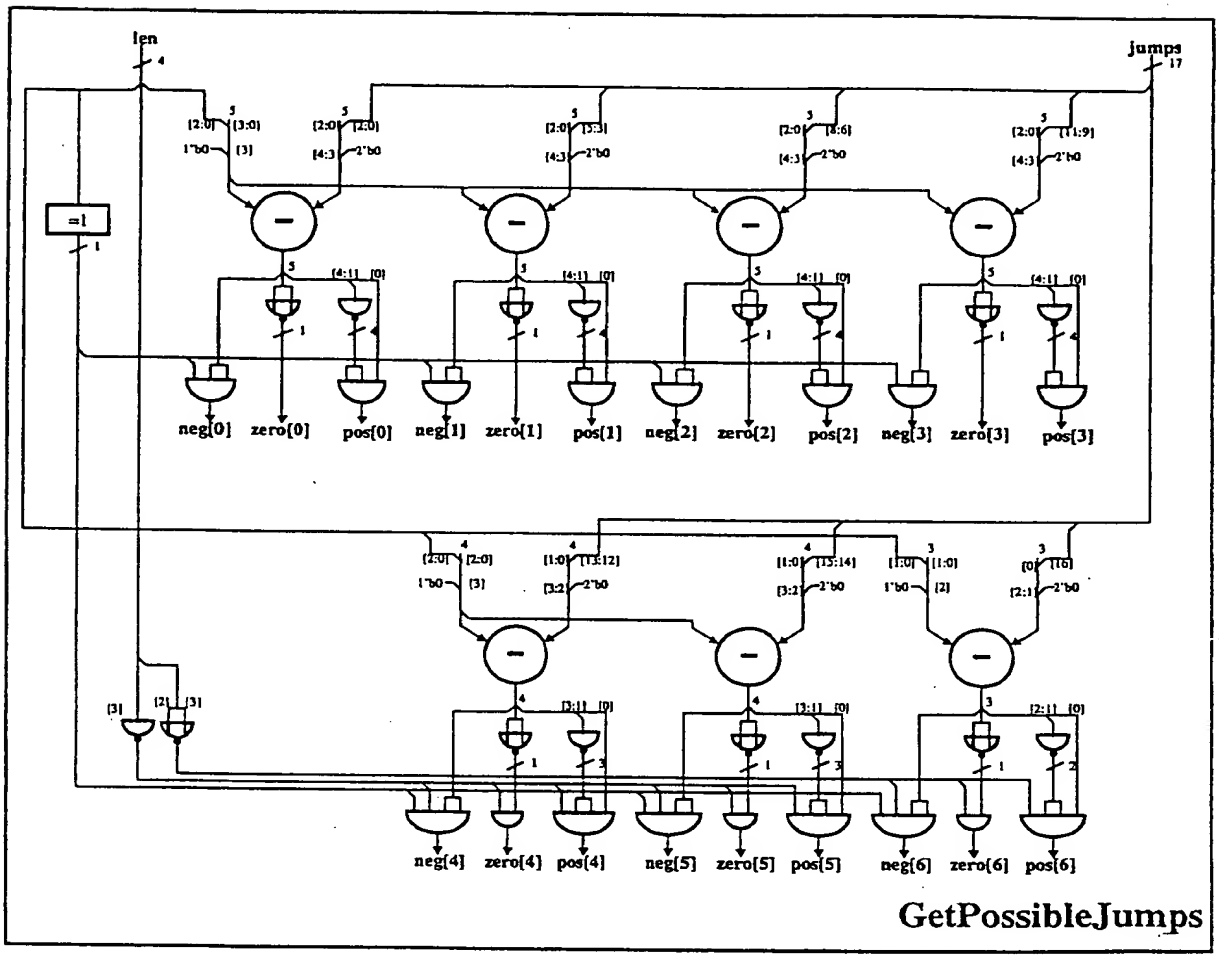

len
jumps
=1
neg[0]
zero[0]
pos[0]
neg[1]
zero[1]
pos[1]
neg[2]
zero[2]
pos[2]
neg[3]
zero[3]
pos[3]
neg[4]
zero[4]
pos[4]
neg[5]
zero[5]
pos[5]
neg[6]
zero[6]
pos[6]
GetPossibleJumps

Fig. 48

Fig. 49

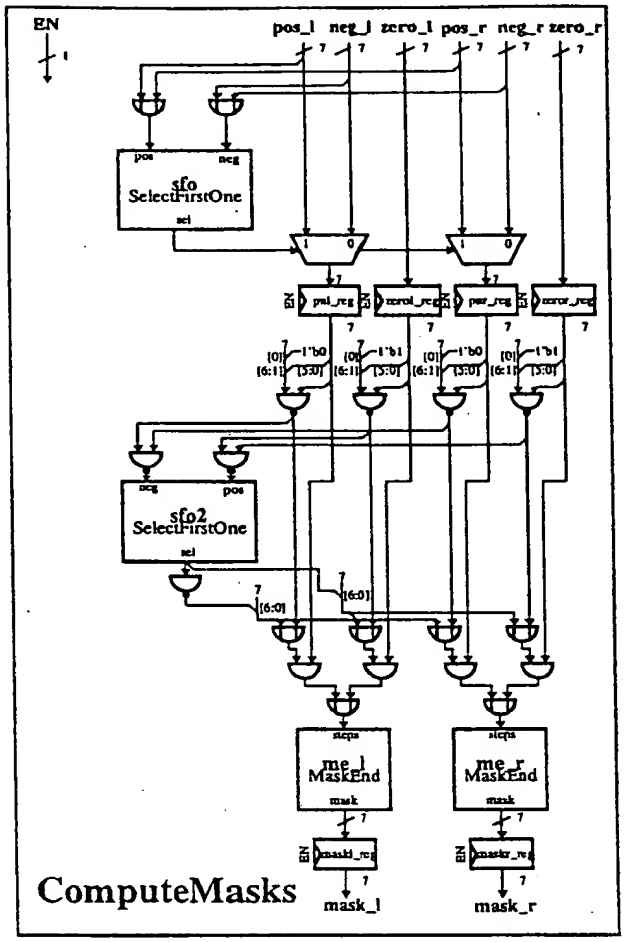
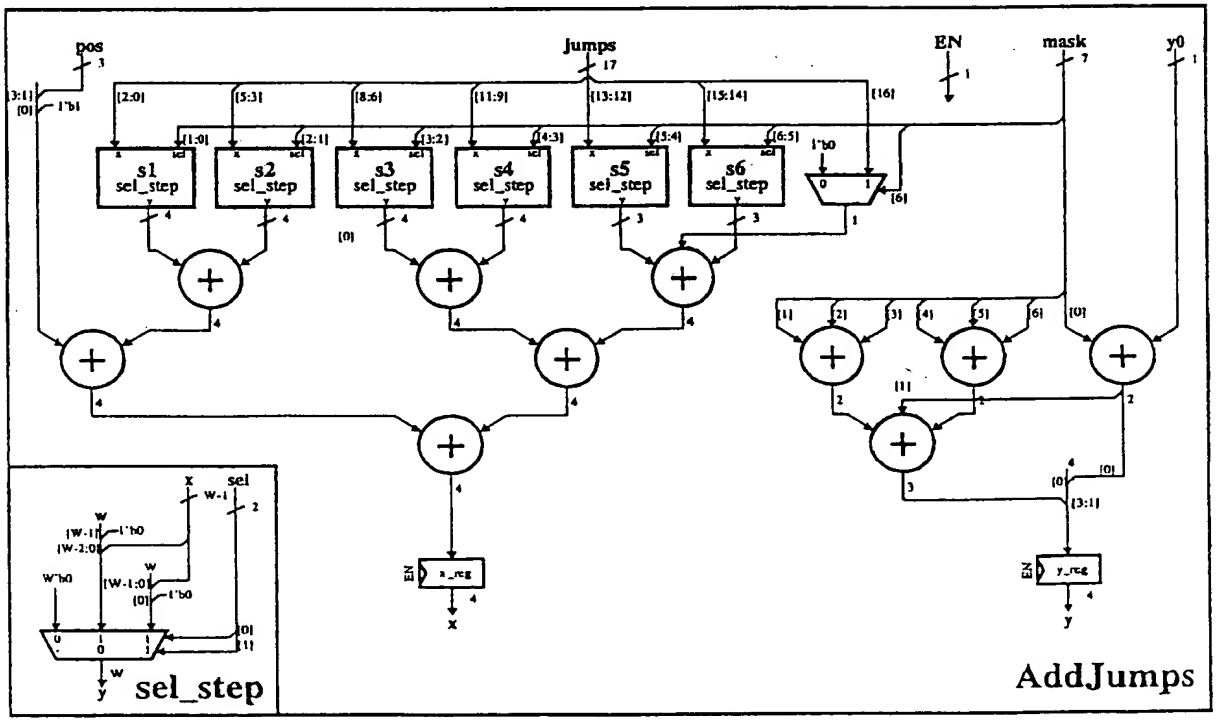

EN
pos_l neg_l zero_l pos_r neg_r zero_r
sfo
SelectFirstOne
sfo2
SelectFirstOne
me_l
MaskEnd
me_r
MaskEnd
mask_l
mask_r
ComputeMasks

Fig. 50


pos
jumps
EN
mask
y0
s1 sel_step
s2 sel_step
s3 sel_step
s4 sel_step
s5 sel_step
s6 sel_step
x
y
sel
w
sel_step
AddJumps

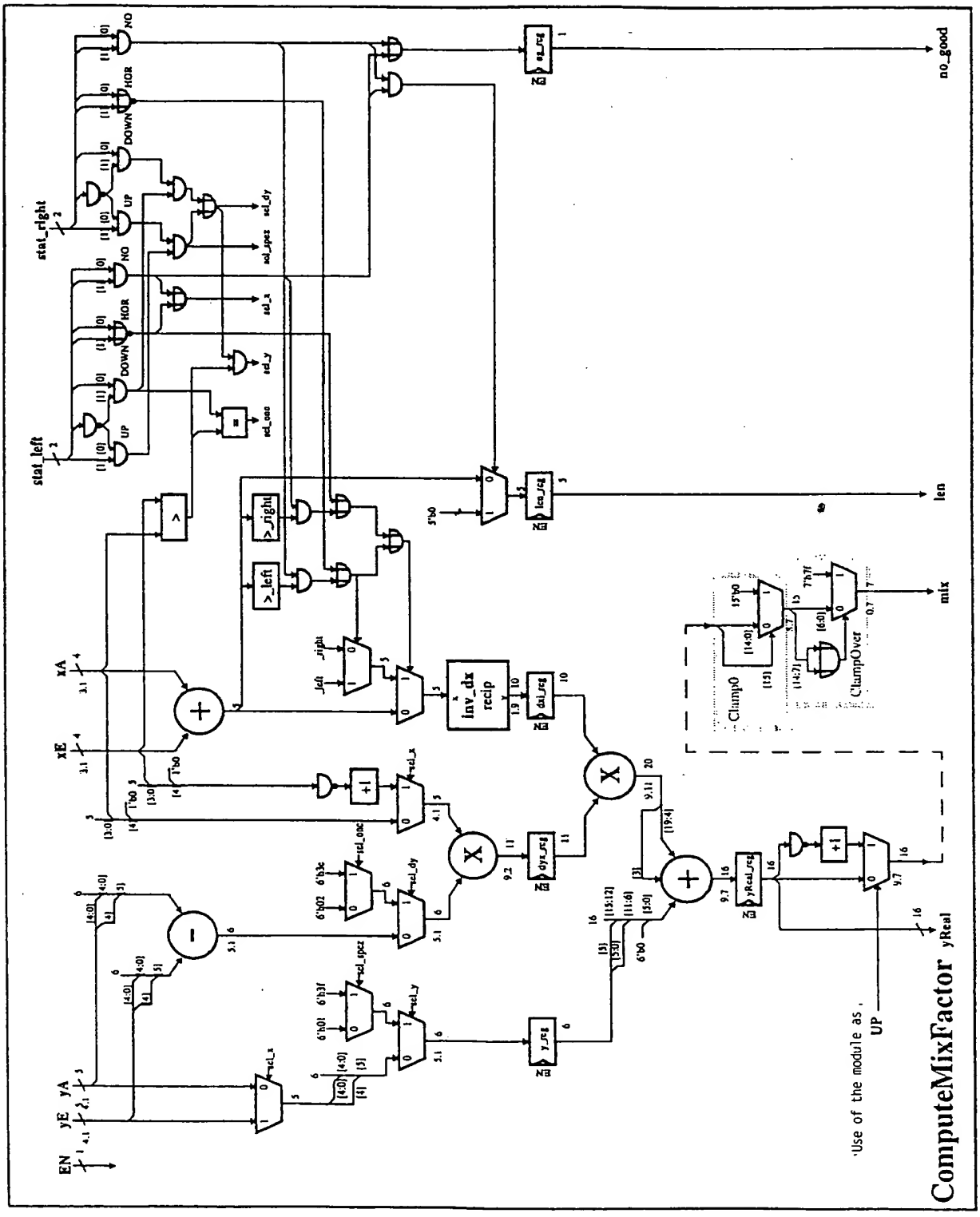

ComputeMixFactor
yReal
mix
len
no_good
stat_right
stat_left
inv_dx recip
ClampOver
Clamp0
'Use of the module as ,
UP

Fig. 51

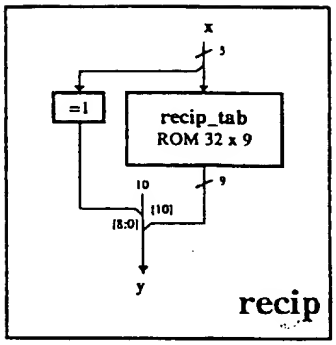


Fig. 52

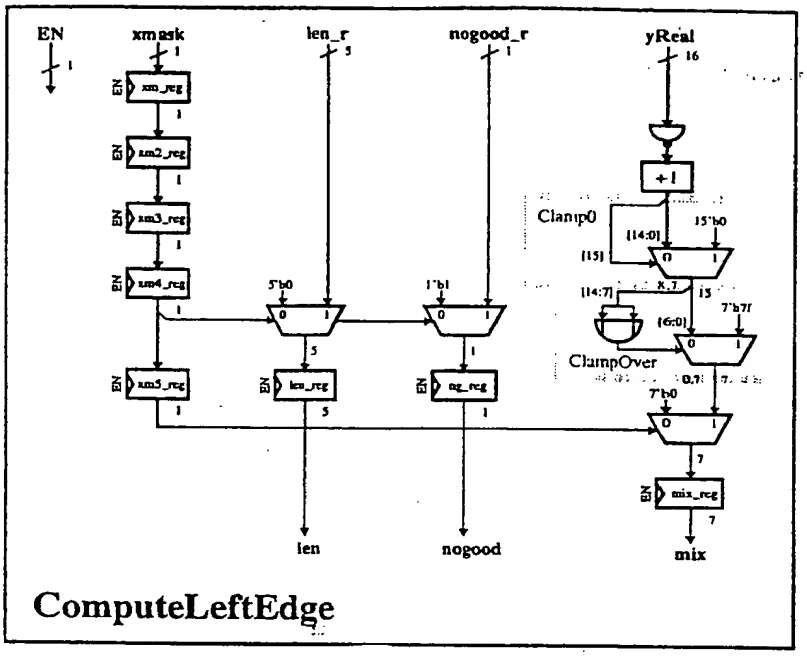


Fig. 53

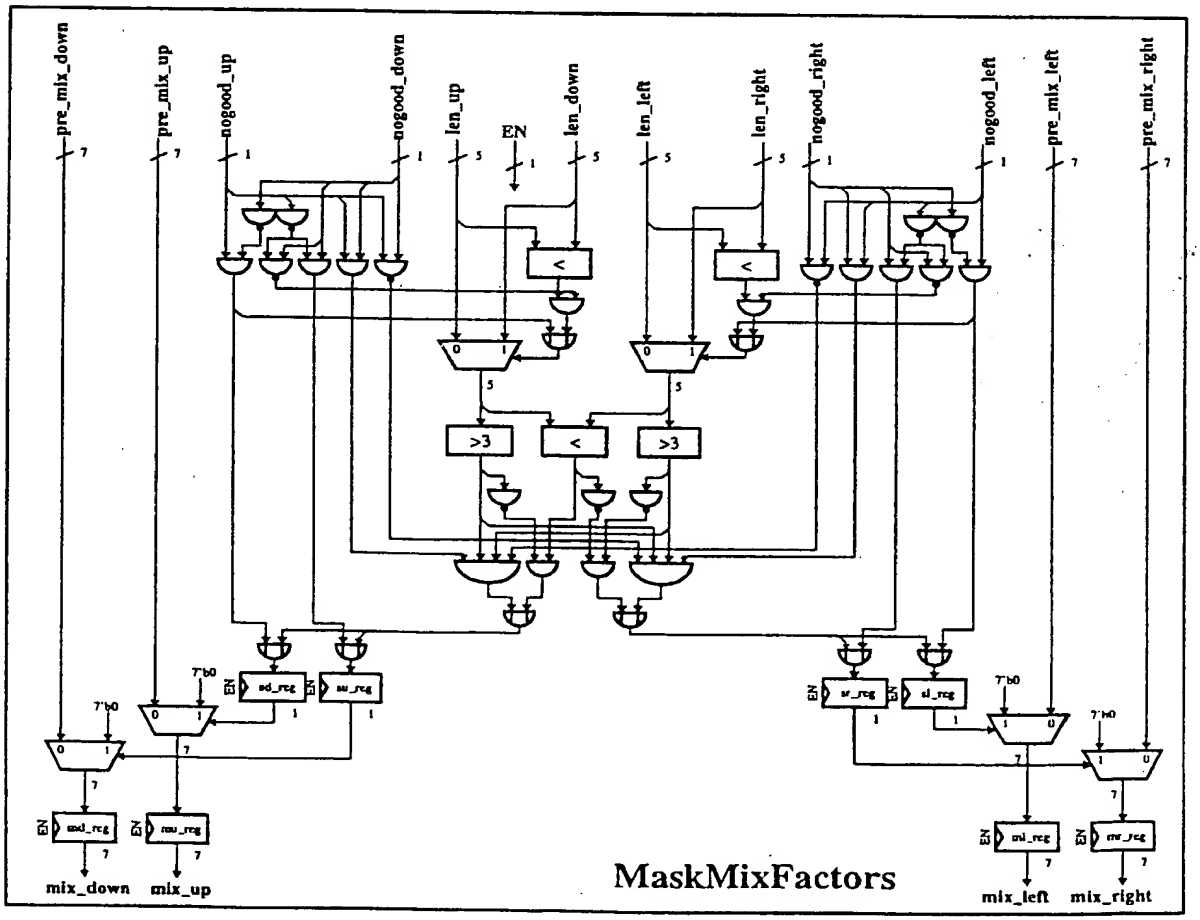


Fig. 54

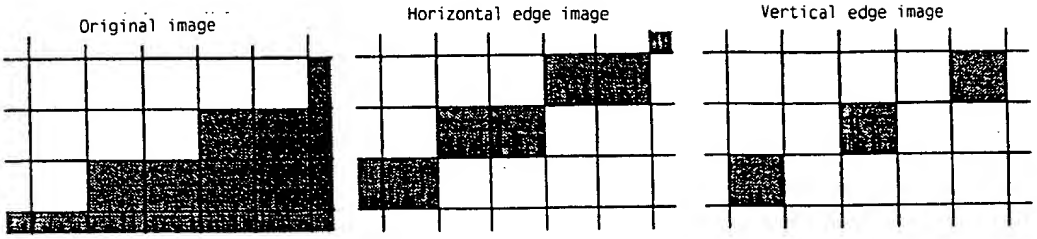

Original image
Horizontal edge image
Vertical edge image

Fig. 55

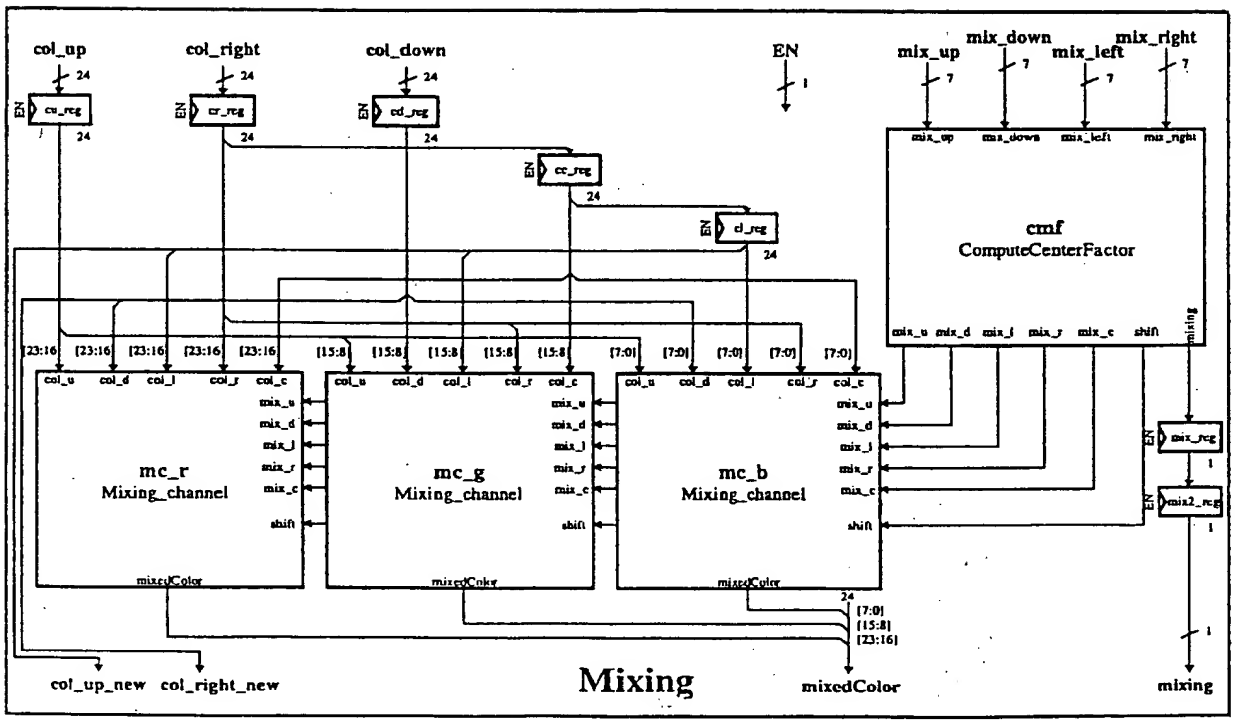

col_up
col_right
col_down
EN
mix_up
mix_down
mix_left
mix_right
cmf
ComputeCenterFactor
mc_r
Mixing_channel
mc_g
Mixing_channel
mc_b
Mixing_channel
col_up_new
col_right_new
Mixing
mixedColor
mixing

Fig. 56

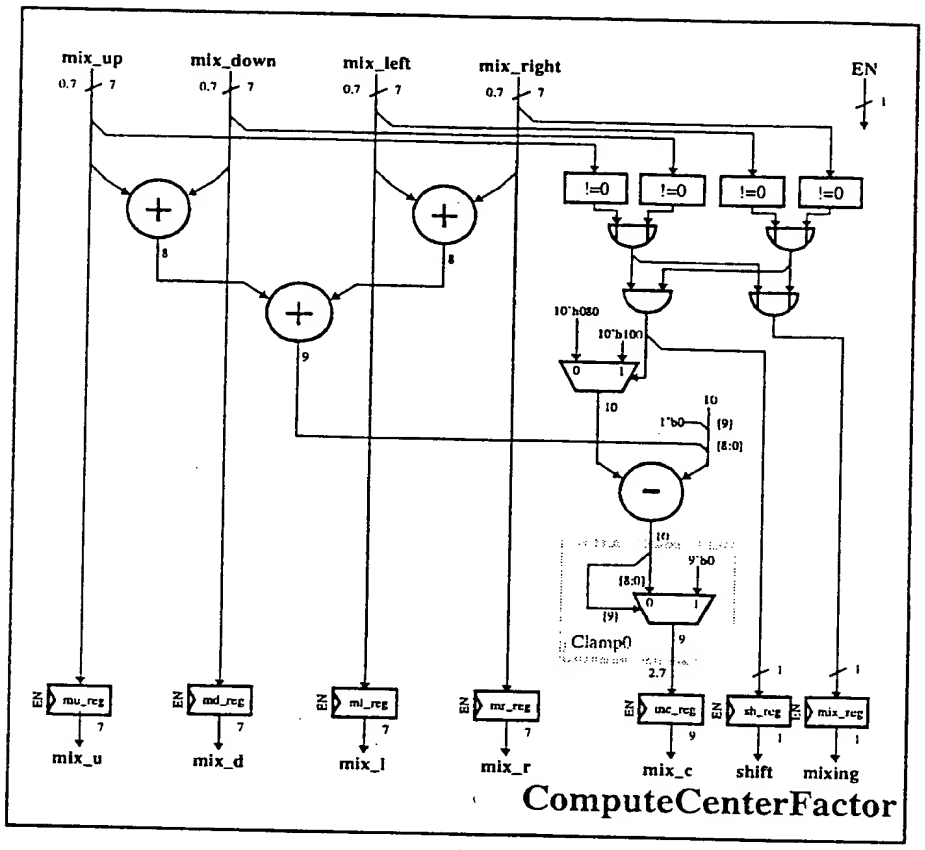

mix_up
mix_down
mix_left
mix_right
EN
!=0
!=0
!=0
!=0
10'b080
10'b100
1'b0
9'b0
Clamp0
mu_reg
md_reg
ml_reg
mr_reg
mc_reg
sh_reg
mix_reg
mix_u
mix_d
mix_l
mix_r
mix_c
shift
mixing
ComputeCenterFactor

Fig. 57

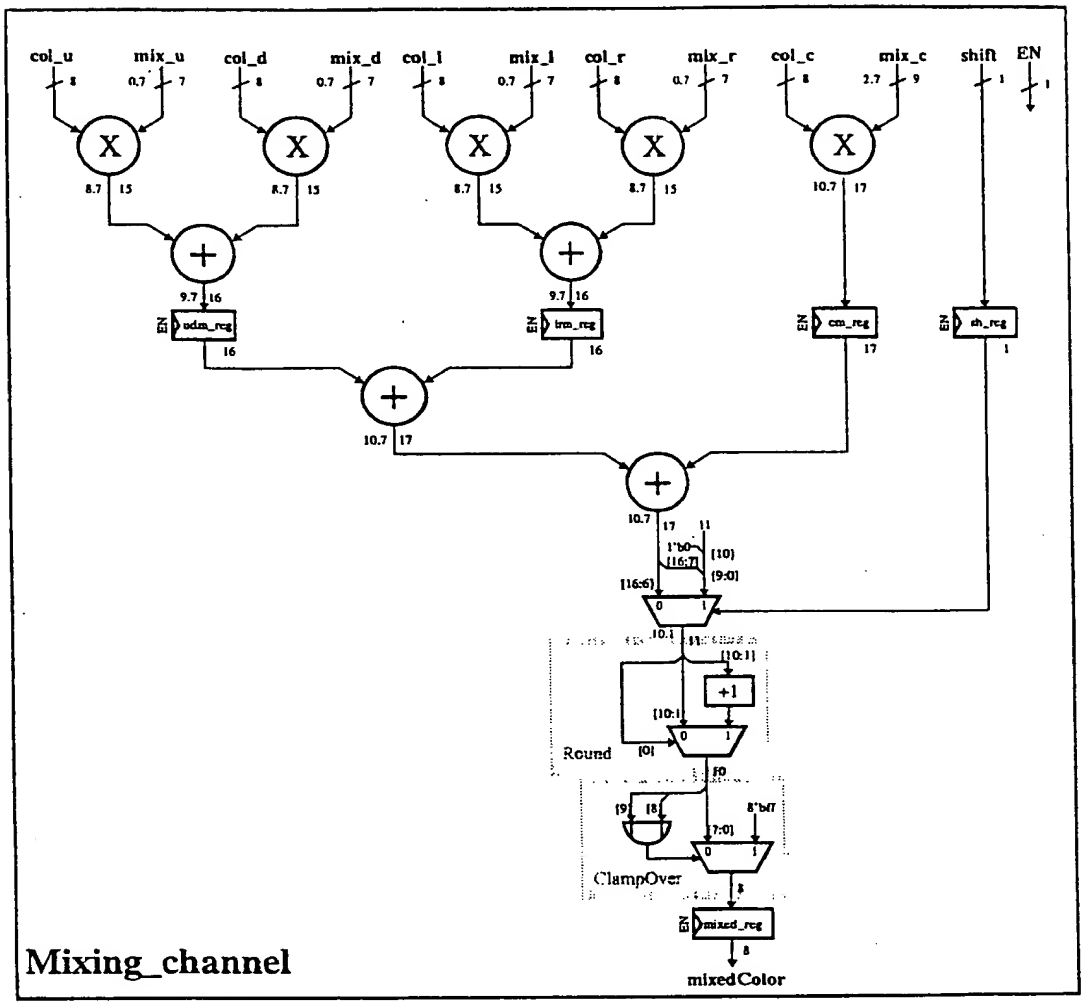


Fig. 58

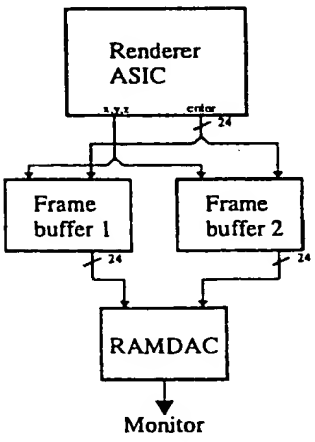


Fig. 59

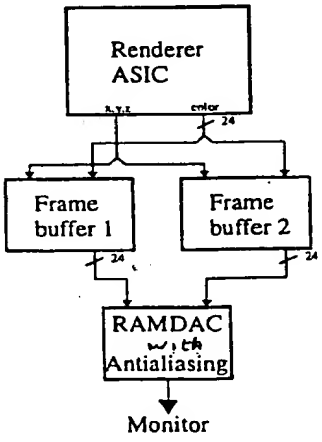


Fig. 60

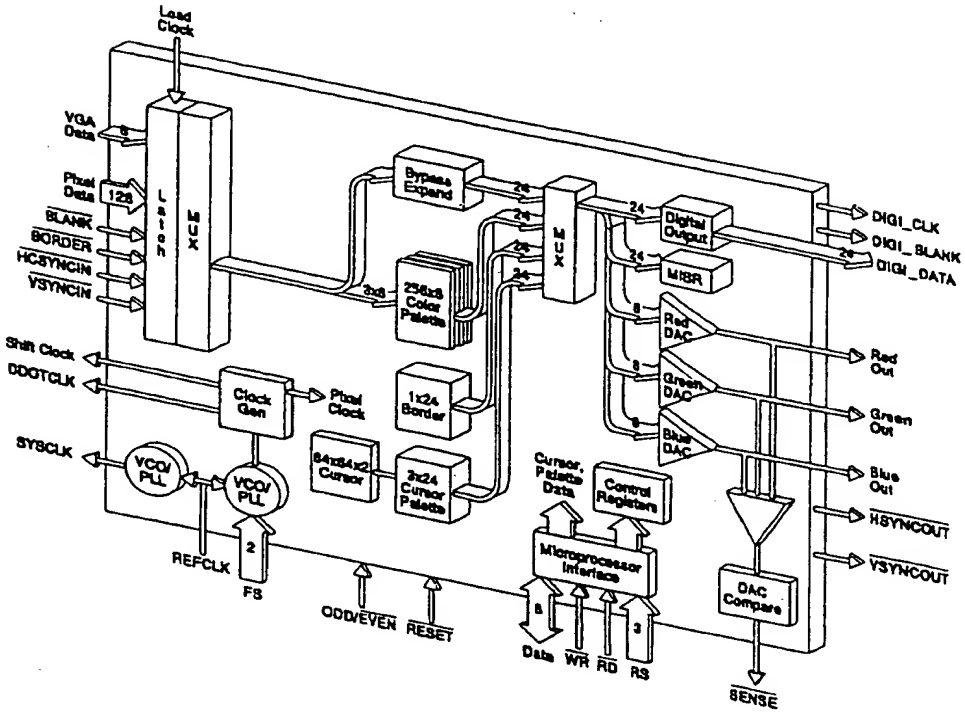
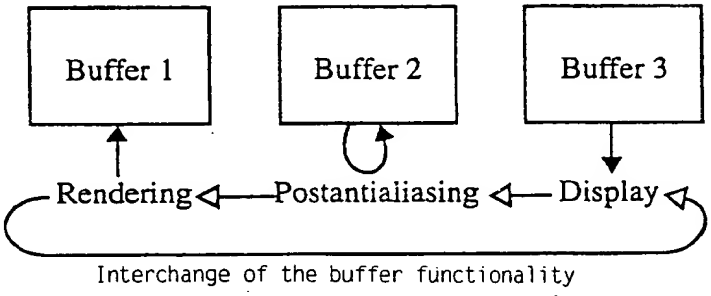


Fig. 61

Fig. 62



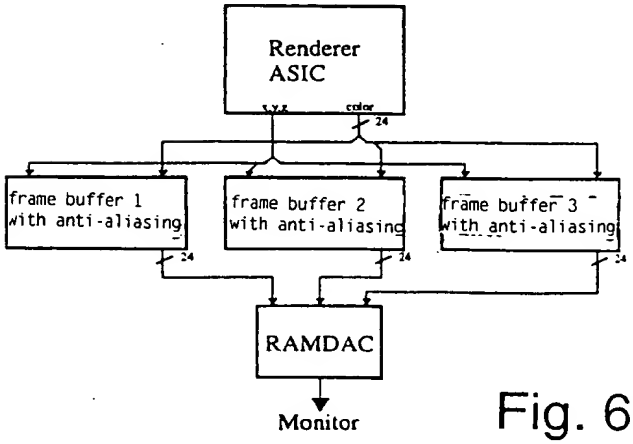


Fig. 63

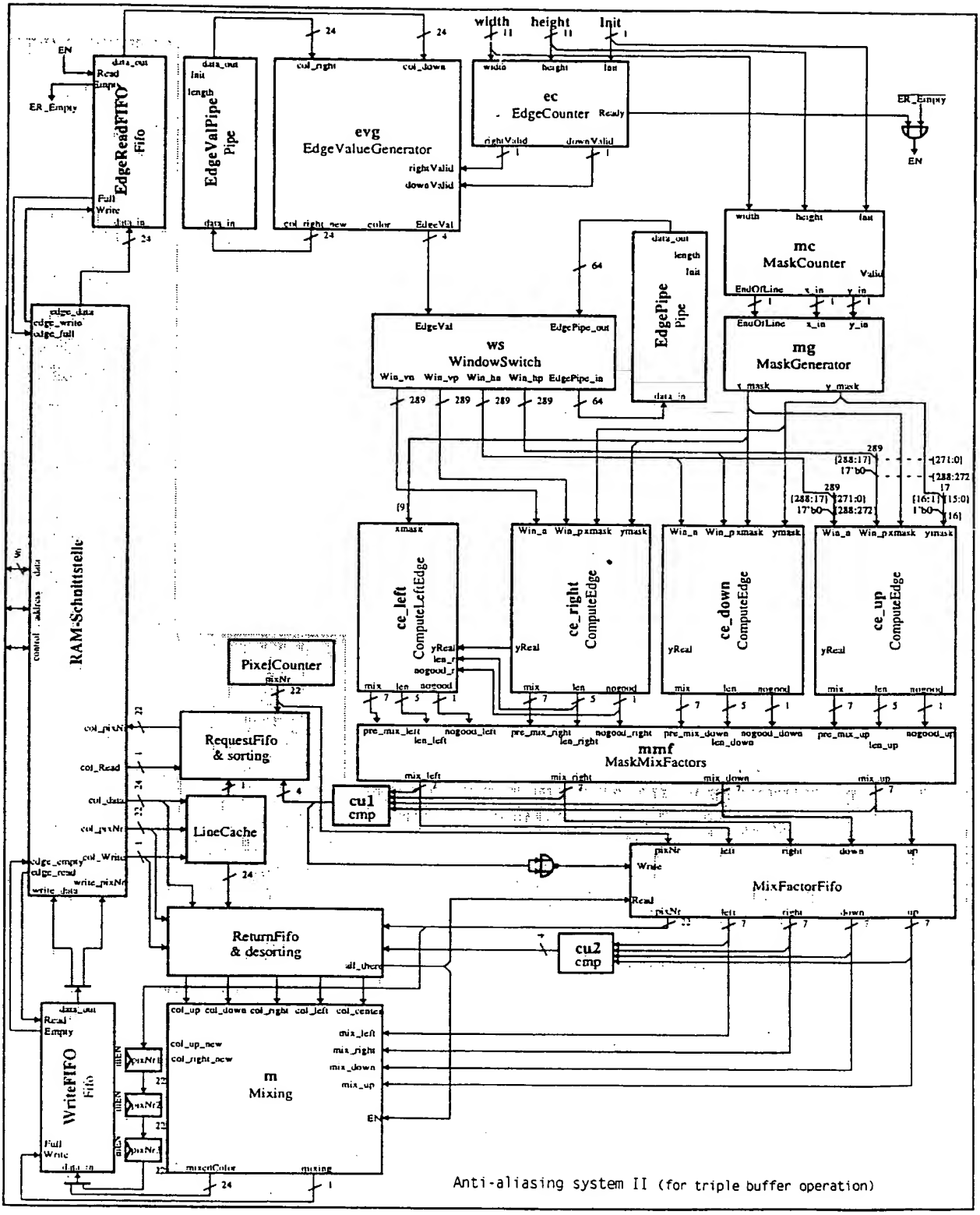

width
height
Init
EdgeReadFIFO
Fifo
EdgeValPipe
Pipe
evg
EdgeValueGenerator
ec
EdgeCounter
ER_Empty
EN
mc
MaskCounter
EdgePipe
Pipe
ws
WindowSwitch
mg
MaskGenerator
RAM-Schnittstelle
PixelCounter
RequestFifo
& sorting
LineCache
ReturnFifo
& desorting
ce_left
ComputeLeftEdge
ce_right
ComputeEdge
ce_down
ComputeEdge
ce_up
ComputeEdge
mmf
MaskMixFactors
cu1
cmp
cu2
cmp
MixFactorFifo
WriteFIFO
Fifo
m
Mixing
Anti-aliasing system II (for triple buffer operation)

Fig. 64

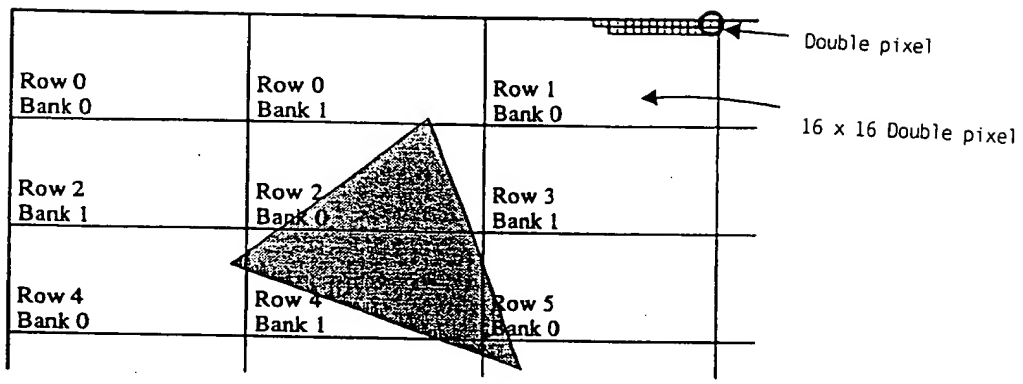

Double pixel
16 x 16 Double pixel
Row 0
Bank 0
Row 0
Bank 1
Row 1
Bank 0
Row 2
Bank 1
Row 2
Bank 0
Row 3
Bank 1
Row 4
Bank 0
Row 4
Bank 1
Row 5
Bank 0

Fig. 65

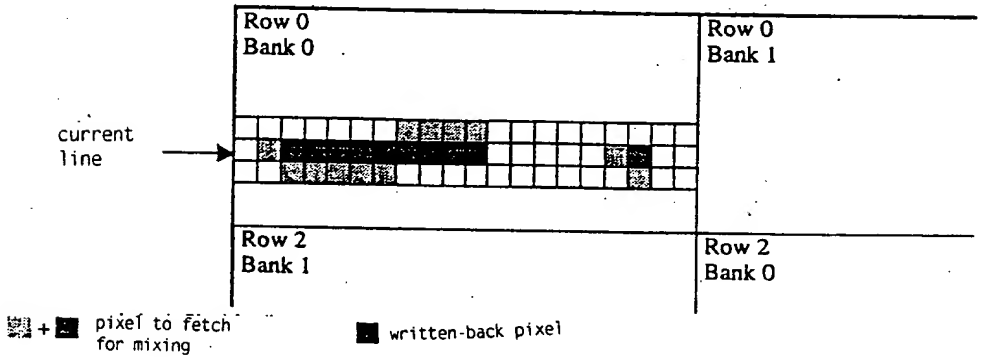
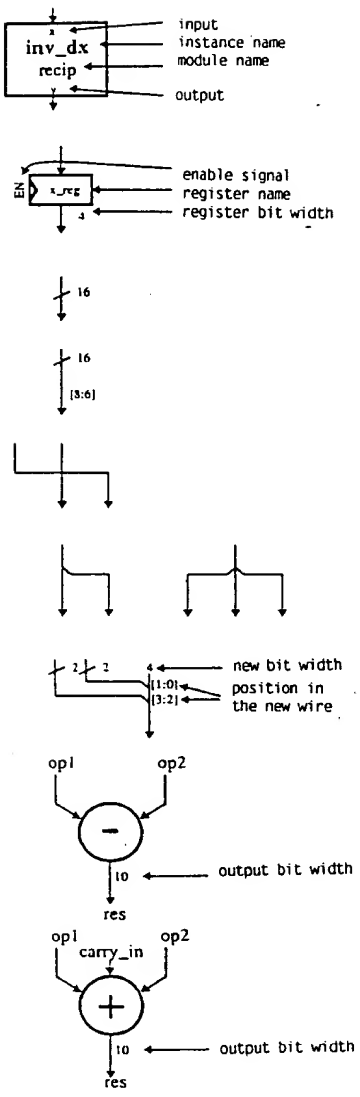

Row 0
Bank 0
Row 0
Bank 1
current
line
Row 2
Bank 1
Row 2
Bank 0
pixel to fetch
for mixing
written-back pixel

Fig. 66



instancing of a self-defined module

register: the datum at the input is transferred at the positive clock edge if the enable signal is set

Bit width of a wire

Selection of bits of a wire

Mutually crossing wires without galvanic connection

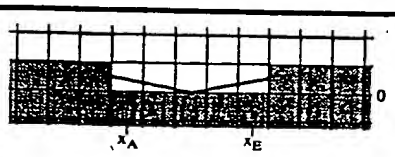
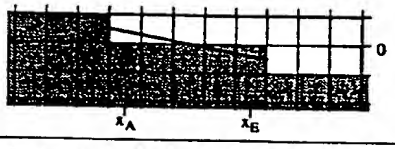
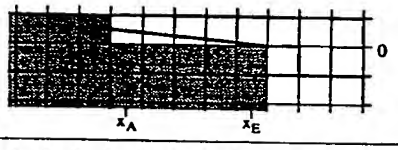
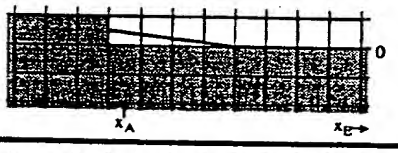
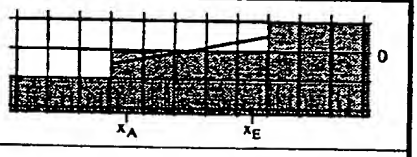
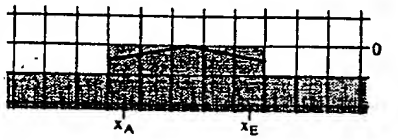
Branching of a wire

Concatenation of bits

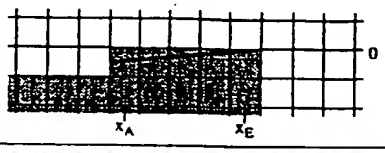
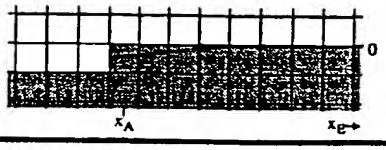
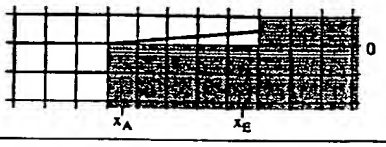
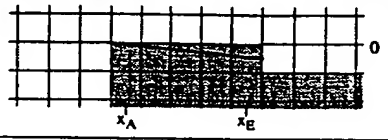
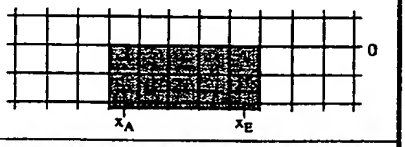
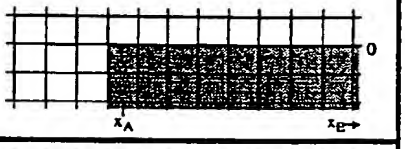
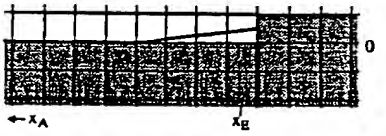
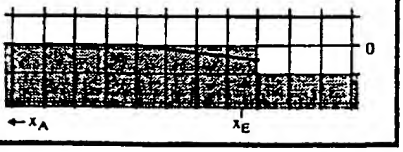
Subtraction (res = op1 - op2/and not the other way round)

Addition (the middle input is only a carry input and thus only 1 bit wide, as cost this element counts as an addition of two numbers)

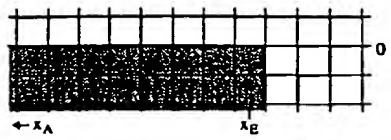
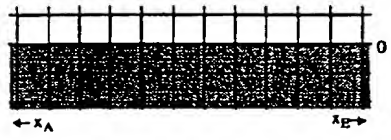
Fig. 67

Start status	End status	Straight line through the points	Example
UP	UP	$(x_A-0.5,0.5)$ - $(0.5*(x_A+x_E),0)$ for $x \leq 0.5*(x_A+x_E)$ $(0.5*(x_A+x_E),0)$ - $(x_E+0.5,0.5)$ for $x > 0.5*(x_A+x_E)$	
UP	DOWN	$(x_A-0.5,0.5)$ - $(x_E+0.5,-0.5)$	
UP	NO	$(x_A-0.5,0.5)$ - $(x_E+0.5,0)$	
UP	HOR	$(x_A-0.5,0.5)$ - $(x_A+0.5*(n-2),0)$	
DOWN	UP	$(x_A-0.5,-0.5)$ - $(x_E+0.5,0.5)$	
DOWN	DOWN	$(x_A-0.5,-0.5)$ - $(0.5*(x_A+x_E),0)$ for $x \leq 0.5*(x_A+x_E)$ $(0.5*(x_A+x_E),0)$ - $(x_E+0.5,-0.5)$ for $x > 0.5*(x_A+x_E)$	

Tab. 1

Start status	End status	Straight line through the points	Example
DOWN	NO	$(x_A\text{-}0.5,\text{-}0.5)$ - $(x_E\text{+}0.5,0)$	
DOWN	HOR	$(x_A\text{-}0.5,\text{-}0.5)$ - $(x_A\text{+}0.5*(n\text{-}2),0)$	
NO	UP	$(x_A\text{-}0.5,0)$ - $(x_E\text{+}0.5,0.5)$	
NO	DOWN	$(x_A\text{-}0.5,0)$ - $(x_E\text{+}0.5,\text{-}0.5)$	
NO	NO	$y \equiv 0$	
NO	HOR	$y \equiv 0$	
HOR	UP	$(x_E\text{-}0.5*(n\text{-}2),0)$ - $(x_E\text{+}0.5,0.5)$	
HOR	DOWN	$(x_E\text{-}0.5*(n\text{-}2),0)$ - $(x_E\text{+}0.5,\text{-}0.5)$	

Tab. 2

Start status	End status	Straight line through the points	Example
HOR	NO	$y \equiv 0$	
HOR	HOR	$y \equiv 0$	

Tab. 3

Signal	Bits	Type	Description
Init	1	In	Initialization signal (transfer of a new height and width)
width	11	In	New display screen width (only relevant at Init=1)
height	11	In	New display screen height (only relevant at Init=1)
Color	24	In	Chromaticity value of a pixel
Valid_in	1	In	Validity display of the color signal
mixedColor	24	Out	Anti-aliased chromaticity value of a pixel
Valid	1	Out	Validity display of the mixed color signal

Tab. 4

Signal	Bits	Type	Description
EN	1	In	Enable signal of the counters
Init	1	In	Initialization signal (transfer of the new width and height)
width	11	In	New display screen width (only relevant at Init=1)
height	11	In	New display screen height (only relevant at Init=1)
rightValid	1	Out	Validity display of the pixel right of the current one
downValid	1	Out	Validity display of the pixel below the current one
Ready	1	Out	Display that all pixels of the current image were read in

Tab. 5

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
col_right	24	In	Chromaticity value for the pixel right of the current one
col_down	24	In	Chromaticity value for the pixel below the current one
rightValid	1	In	Validity display of the col_right signal
downValid	1	In	Validity display of the col_down signal
EdgeVal	4	Out	Edge information for the current pixel
col_right_new	24	Out	Chromaticity value which is used again for edge generation in the next line
color	24	Out	Chromaticity value which is to be stored for later mixing

Tab. 6

Signal	Bits	Type	Description
EN	1	In	Enable signal of the module
color0	24	In	Color of the one pixel
color1	24	In	Color of the other pixel between which an edge is to be recognised
cmp	1	Out	Indication whether an edge exists between the pixels involved

Tab. 7

Signal	Bits	Type	Description
EN	1	In	Enable signal of the module
r_color	24	In	Color of the right pixel
c_color	24	In	Color of the central pixel
d_color	24	In	Color of the lower pixel
r_pos	1	Out	There is a positive color jump from the central to the right pixel
r_neg	1	Out	There is a negative color jump from the central to the right pixel
d_pos	1	Out	There is a positive color jump from the central to the lower pixel
d_neg	1	Out	There is a negative color jump from the central to the lower pixel

Tab. 8

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
x	8	In	Chromaticity value of a channel
y	8	In	Chromaticity value of a channel
z	8	In	Chromaticity value of a channel
approx	11	Out	Result value

Tab. 9

Signal	Bits	Type	Description
EN	1	In	Enable signal of the module
EdgeVal	4	In	New edge value for the right lower pixel in the edge window
EdgePipe_out	64	In	Edge values of the most right column
Win_vp	289	Out	Edge window with the vertically positively marked pixels
Win_vn	289	Out	Edge window with the vertically negatively marked pixels
Win_hp	289	Out	Edge window with the horizontally positively marked pixels
Win_hn	289	Out	Edge window with the horizontally negatively marked pixels
EdgePipe_in	64	Out	Edge values of the column most to the left, which are no longer required for the current window relevant again only when processing the next line)

Tab. 10

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
EdgeVal	1	In	New edge value for the right lowermost pixel
EdgePipe_out	16	In	Edge value for the most right column
actWin	289	Out	Vertical/horizontal edge window
EdgePipe_in	16	Out	Edge values of the column most to the left, which are no longer required for the current window (relevant again only when processing the next line)

Tab. 11

Signal	Bits	Type	Description
Reset	1	In	Reset signal. after application of this signal an RAM initialization phase occurs
Init	1	In	Initialization signal at which the length of the pipe is established again
L	?	In	New length of the pipe used (bit width can be established by way of parameters)
Shift	1	In	Signal for advancing the data (new datum is taken over, a datum is outputted)
DIn	?	In	Data input of the pipe (bit width can be established by way of parameters)
DOut	?	Out	Data output of the pipe (bit width can be established by way of parameters)

Tab. 12

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
Init	1	In	Initialization signals for the masks
x_in	1	In	New value for the x-mask
y_in	1	In	New value for the y-mask
EndOfLine	1	In	Signal which displays that the end of an image line is reached
xmask	17	Out	Mask which horizontally specifies which pixels belong to the current environment
ymask	17	Out	Mask which vertically specifies which pixels belong to the current environment

Tab. 13

Signal	Bits	Type	Description
EN	1	In	Enable signal of the counters
Init	1	In	Initialization signal (transfer of the new width and height)
width	11	In	New width (only relevant at Init=1)
height	11	In	New height (only relevant at Init=1)
EndOfLine	1	Out	Signal which indicates that the end of an image line is reached
x_in	1	Out	New value for the x-mask
y_in	1	Out	New value for the y-mask
Valid	1	Out	Is the current pixel valid at all (within the image limits)?

Tab. 14

Signal	Bits	Type	Description
EN	1	In	Enable signal of the module
Win_p	289	In	Positively marked pixel in the edge window
Win_n	289	In	Negatively marked pixel in the edge window
xmask	17	In	x-mask (validity of the pixels in the edge window)
ymask	17	In	y-mask (validity of the pixels in the edge window)
yReal	16	Out	Position of the straight line with respect to the color jump
mix	7	Out	Mix factor which specifies how much of the color from the corresponding direction is to be mixed (if only that straight line exists)
len	5	Out	Horizontal length of the traced straight line
nogood	1	Out	Flag which specifies whether the straight line is deemed "good"

Tab. 15

Signal	Bits	Type	Description
actWin_p	289	In	Positively marked pixels in the edge window
actWin_n	289	In	Negatively marked pixels in the edge window
xmask	17	In	Validity of the pixels in the window
ymask	17	In	Validity of the lines in the window
maskedWindow	191	Out	Masked edge window

Tab. 16

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
ep	191	In	Edge window (edge pic/no longer rectangular)
stat_left	2	Out	Status at the left end of the traced step
stat_right	2	Out	Status at the right end of the traced step
xA	4	Out	x-value of the start point (actually -xA but sign is not required/value range:0..7.5)
yA	5	Out	y-value of the start point (value range:-7.5..+7.5)
xE	4	Out	x-value of the end point (value range:0..7.5)
yE	5	Out	y-value of the end point (value range: -7.5..+7.5)

Tab. 17

Signal	Bits	Type	Description
up	9	In	Line in the edge image above the middle one
center	9	In	Middle line in the edge image in one direction
down	9	In	Line in the edge image below the middle one
len	3	Out	Length of the central step in the traced direction
stat	3	Out	Status at the end of the central step

Tab. 18

Signal	Bits	Type	Description
tstat_l	3	In	Temporary status towards the left
tstat_r	3	In	Temporary status towards the right
stat_l	2	Out	Status at the left end of the traced step
stat_r	2	Out	Status at the right end of the traced step

Tab. 19

tstat	Bits[2:0]
UP	010
DOWN	001
NO	000
HOR	011
BOTH	100

Tab. 20

stat	Bits[1:0]
UP	10
DOWN	01
NO	00
HOR	11

Tab. 21

Signal	Bits	Type	Description
pos	3	In	Position at which the central step ends
ep	44	In	Sector from the edge image in which further steps are to be found
jumps	17	Out	Jumps found in the sector

Tab. 22

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
stat_l	2	In	Status at the left edge of the central step
stat_r	2	In	Status at the right edge of the central step
pos_l	3	In	Length of the central step towards the left
pos_r	3	In	Length of the central step towards the right
jumps_l	17	In	Further steps recognised towards the left
jumps_r	17	In	Further steps recognised towards the right
xA	4	Out	x-value of the start point
yA	4	Out	y-value of the start point (without sign)
xE	4	Out	x-value of the end point
yE	4	Out	y-value of the end point (without sign)

Tab. 23

Signal	Bits	Type	Description
len	4	In	Length of the central step
jumps	17	In	Possible jumps
pos	7	Out	Mask which specifies which steps include a length greater by one than the central step
zero	7	Out	Mask which specifies which steps contain the same length as the central step
neg	7	Out	Mask which specifies which steps contain a length shorter by one than the central step

Tab. 24

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
pos_l	7	In	Mask which contains the steps longer by one towards the left
zero_l	7	In	Mask which contains the steps of the same length towards the left
neg_l	7	In	Mask which contains the steps shorter by one towards the left
pos_r	7	In	Mask which contains the steps longer by one towards the right
zero_r	7	In	Mask which contains steps of the same length towards the right
neg_r	7	In	Mask which contains the steps shorter by one towards the right
mask_l	7	Out	Mask which specifies which steps towards the left contribute to a straight line
mask_r	7	Out	Mask which specifies which steps towards the right contribute to a straight line

Tab. 25

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
pos	3	In	x-value of the end of the central step
y0	1	In	y-value of the end of the central step
jumps	17	In	Further steps in the corresponding direction
mask	7	In	Mask which specifies which of the steps in jumps contribute to the straight line
x	4	Out	x-value of the one end point of the straight line
y	4	Out	Magnitude of the y-value of the one end point of the straight line

Tab. 26

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
xA	4	In	x-value of the start point
yA	5	In	y-value of the start point
xE	4	In	x-value of the end point
yE	5	In	y-value of the end point
stat_left	2	In	Status at the start point
stat_right	2	In	Status at the end point
yReal	16	Out	y-value of the real straight line in the current pixel
mix	7	Out	Mixed factor for a mixing direction rel. to the straight line
len	5	Out	Length of the traced straight line
no_good	1	Out	Statuses at the ends of the straight line indicate not a "benign" straight line

Tab. 27

Signal	Bits	Type	Description
x	5	In	Input value
y	10	Out	Input value = $\frac{1}{x}$

Tab. 28

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
xmask	1	In	Mask bit which specifies whether the current pixel belongs to an edge
yReal	16	In	y-value of the real edge relative to the color jump
len_r	5	In	Length of the traced straight line for the right mix factor
nogood_r	1	In	Flag indicating that the straight line was probably not "benign" for ascertaining the right mix factor
mix	7	Out	Left mix factor for the current pixel
len	5	Out	Length of the traced straight line
nogood	1	Out	Statuses at the end of the straight line indicate not a "benign" straight line

Tab. 29

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
pre_mix_up, pre_mix_down, pre_mix_left, pre_mix_right	je 7	In	Mix factors which were established on the basis of only one respective straight line
len_up, len_down, len_left, len_right	je 5	In	Lengths of the respective straight line
nogood_up, nogood_down, nogood_left, nogood_right	je 1	In	Flags of the respective straight lines which cannot be deemed "good"
mix_up, mix_down, mix_left, mix_right	je 7	Out	Definitive mixing factors for the corresponding directions

Tab. 30

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
mix_up, mix_down, mix_left, mix_right	je 7	In	Mix factors for the corresponding directions.
col_up	24	In	Color of the pixel above the current one
col_right	24	In	Color of the pixel to the right of the current one
col_down	24	In	Color of the pixel below the current one
mixedColor	24	Out	New color of the current pixel by virtue of mixing
mixing	1	Out	Flag whether mixing occurred (in triple buffer mode the decision whether the pixel has to be written back)
col_up_new	24	Out	Color which is used in the next line as uppermost pixel
col_right_new	24	Out	Color which is used in the next line as right pixel

Tab. 31

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
mix_up, mix_down, mix_left, mix_right	je 7	In	Mix factors for the corresponding directions
mix_u, mix_d, mix_l, mix_r	je 7	Out	Mix factors delayed by one cycle
mix_c	7	Out	Central mix factor
shift	1	Out	Flag whether the mixed factors must still all be divided by two
mixing	1	Out	Flag whether mixing occurs or whether the central factor is 1.0

Tab. 32

Signal	Bits	Type	Description
EN	1	In	Enable signal of the registers
mix_u, mix_d, mix_l, mix_r, mix_c	je 7	In	Mix factors for the corresponding pixels
col_u, col_d, col_l, col_r, col_c	je 8	In	Colors of the individual pixels in the respective color channel
shift	1	In	Flag that division must still occur by 2
mixedColor	8	Out	Mixed color for the color channel

Tab. 33